UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

INSTITUTO DE INFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Gerenciamento Integrado de QoS em Redes de Computadores

por

LISANDRO ZAMBENEDETTI GRANVILLE

Tese submetida à avaliação, como requisito
parcial para obtenção do grau de Doutor
em Ciência da Computação

Profa. Dra. Liane Margarida Rockenbach Tarouco

Orientadora

Porto Alegre, setembro de 2001

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Granville, Lisandro Zambenedetti

Gerenciamento Integrado de QoS em Redes de Computadores / por Lisandro Zambenedetti Granville. – Porto Alegre: PPGC da UFRGS, 2000.

159p.: il.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, BR - RS, 2001. Orientador: Tarouco, Liane M. R.

1. Redes de Computadores. 2. Gerência de Redes. 3. Qualidade de Serviço. 4. Gerenciamento Integrado. I. Tarouco, Liane Margarida Rockenbach. II. Título.

Agradecimentos

Inicialmente gostaria de agradecer à Profa. Liane Tarouco pelo incentivo e orientação. As idéias discutidas durante estes anos foram imprescindíveis para o desenvolvimento deste trabalho.

Quero agradecer também aos professores do instituto de informática da UFRGS, principalmente ao professores do grupo de redes, pelo apoio na realização deste trabalho, seja através da disponibilização de equipamentos para os testes, seja através das longas discussões.

Aos colegas de pós-graduação Luciano Gaspary, André Nácul, Gustavo Coelho, Marcelo Ribeiro e Márcio Ceccon pelas inúmeras horas de discussão, e pelas incansáveis corridas para cumprir os prazos de submissão de artigos.

À minha namorada, Daniela Leal Musa, pelo enorme apoio, compreensão, amor e carinho, sem os quais eu não teria conseguido terminar esta tese. Por fim, aos meus pais Maria e José, que muito têm me incentivando e apoiando durante toda minha caminhada, e aos meus irmãos Luciane e Leonardo pelo companheirismo e amizade.

Sumário

Lista de Abreviaturas

API	Application Program Interface
ATM	Asynchronous Transfer Mode
BB	Bandwidth Broker
CLI	Command Line Interface
COPS	Common Open Policy Service
CORBA	Common Object Request Broker Architecture
CQ	Custom Queuing
DiffServ	Differentiated Services
DMTF	Desktop Management Task Force
DNS	Domain Name Server
DS	DiffServ
EEM	ExtremeAware Enterprize Manager
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
HDR	High Data Rate
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IntServ	Integrated Services
IP	Internet Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MIB	Management Information Base
MIB-II	Management Information Base version II
MLM	Mid-Level Manager
MPLS	Multi Protocol Label Switching
MRTG	Multi Router Traffic Grapher
NS	Network Simulator
OSI	Open Systems Interconnection
PBNM	Policy-Based Network Management
PDP	Policy Enforcement Point
PDV	Ponto De Venda
PEP	Policy Enforcement Point
PHP	PHP: Hypertext Processor
PIB	Policy Information Base
PQ	Priority Queuing
QAME	QoS-Aware Management Environment
QPM	QoS Policy Manager
QoS	Quality of Service
RAP	Resource Allocation Protocol
RFC	Request For Comments

RMON	Remote Monitoring
RMON2	RMON version 2
RSVP	Resource Reservation Protocol
RTCP	Real-Time Control Protocol
RTP	Real-Time Protocol
SAP	Service Access Point
SBM	Subnet Bandwidth Management
SDL	Specification Description Language
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SNMPv2	SNMP version 2
SNMPv3	SNMP version 3
SQL	Structured Query Language
SSH	Secure Shell
TCP	Transmission Control Protocol
ToS	Type of Service
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VoIP	Voice over IP
WAN	Wide Area Network
WBEM	Web-Based Enterprise Management
WFQ	Weighted Fair Queuing
XML	Extensible Markup Language

Lista de Figuras

Lista de Tabelas

Resumo

O gerenciamento de redes de computadores é uma tarefa complexa de ser realizada porque a redes atualmente possuem muitos equipamentos, protocolos, serviços e usuários. Como regra geral, redes de computadores são heterogêneas e a introdução de facilidades para o fornecimento de qualidade de serviço (QoS) faz com que a gerência de redes se torne ainda mais complexa, e neste cenário as soluções tradicionais de gerenciamento podem falhar.

O gerenciamento de redes com QoS vem sendo investigado por diversos grupos, e soluções comerciais para o problema já podem ser encontradas. Entretanto, a gerência de QoS possui aspectos diversos que são investigados separadamente. Do ponto de vista do administrador de redes, soluções pontuais são importantes, mas a integração entre os diversos aspectos de gerência de QoS também é uma necessidade. Tal necessidade, entretanto, não tem encontrado respaldo nas pesquisas desenvolvidas ou nos produtos de gerenciamento disponibilizados no mercado. Nesta situação, o administrador da rede é obrigado a trabalhar ao mesmo tempo com soluções disjuntas que não oferecem integração. Isso acaba por complicar a gerência de QoS, que por si só já é uma atividade complexa.

Nesta tese de doutorado serão verificadas as problemáticas associadas à gerência de QoS e discutida a necessidade de integração. As soluções existentes e pesquisas desenvolvidas são classificadas para que possam ser mais bem avaliadas em uma visão mais global. A classificação realizada organiza os aspectos de QoS em tarefas de gerência de QoS que devem ser executadas por um administrador de redes que possuam facilidades de QoS.

Para que a integração das tarefas de gerência de QoS seja possível, um modelo para gerência integrada de QoS em redes de computadores é proposto. O modelo, definido em camadas, utiliza elementos distribuídos na rede para executar tarefas específicas e um ambiente de gerência central fornece uma interface única de acesso ao sistema de gerência ao administrador da rede.

O modelo proposto é analisado de acordo com alguns aspectos (por exemplo, em relação à sua escalabilidade, flexibilidade e exeqüibilidade). Uma implementação do mesmo é apresentada para a gerência de redes baseadas em IP que possuam mecanismos para fornecimento de QoS. A avaliação do modelo mostra, ao final, que a gerência de QoS, além de ser uma necessidade real, é possível de ser executada de forma integrada, como desejam os administradores de rede.

Palavras-chave: redes de computadores, gerência de redes, qualidade de serviço, gerenciamento integrado.

Title: “Integrated Management of QoS in Computer Networks”

Abstract

The management of modern computer networks is a complex task because current networks are typically composed by several devices, protocols, services and users. As a general rule, computer networks have a heterogeneous nature and the introduction of facilities to provide quality of service (QoS) brings new complexities to the already complex environment. In this scenario, traditional network management solutions may fail.

The management of QoS-enabled networks has been investigated by several research groups, and commercial solutions can be already found on the market. However, the QoS management has such diverse aspects that are separately investigated. From a network administrator’s point of view, specific solutions are important, but the integration of the several aspects involved in the whole QoS issue is also a necessity. This necessity, on the other hand, does not find support neither in the developed researches nor in the management tools available on the market. In this situation, the network administrator is forced to deal with disjoint solutions at the same time, with no integration among them. This turns the management of QoS aspects, which is complex by itself, ever more complex.

In this PhD. thesis the problems associated to the management of QoS-enabled networks will be investigated, as long as the necessity of integration. The available solutions and developed researches will be classified to be better evaluated in a more global view. Such classification organizes the aspects related to QoS in QoS management-related tasks that must be executed by the administrator of networks that present QoS facilities.

To allow the integration of such QoS management-related tasks, a model for integrated management of QoS in computer networks is proposed. This model owns elements, organized in layers, that are distributed all over the managed network and are supposed to execute management specific tasks. The model uses a centralized management environment to provide a single management access interface to the network administrator.

The proposed model is verified according to some important aspect (for example, related to its scalability, flexibility and exequibility). An implementation of the model is also presented. This implementation is used for the management of IP-based networks with QoS provisioning mechanisms. The analysis of such model shows, at the end, that the management of QoS, besides being a real necessity, is possible to be executed in an integrated fashion.

Keywords: computer networks, network management, quality of service, integrated management.

1 Introdução

O objetivo principal da gerência de redes de computadores é manter em funcionamento a estrutura de comunicação utilizada pelos usuários das redes. O nível mais básico da gerência se preocupa com a transmissão de dados pura. O nível mais especializado se preocupa com o adequado funcionamento dos serviços oferecidos aos usuários.

Atualmente, a maior parte das redes instaladas opera no paradigma de melhor esforço (*best-effort*), onde não existe nenhuma garantia sobre a qualidade do transporte de dados. Em várias situações diferentes as informações transmitidas podem ser perdidas, enfrentarem atrasos não esperados, possuírem diferenças sensíveis nesses atrasos, além de não contarem com uma largura de banda garantida. O paradigma de melhor esforço pode ser suficiente para várias aplicações, como transferência de arquivos, navegação Web e correio eletrônico. Entretanto, várias outras aplicações simplesmente não podem operar adequadamente em tal ambiente. Aplicações de videoconferência, telemedicina e ensino à distância, por exemplo, necessitam de certas garantias dos serviços de rede que não são fornecidas pelo serviço de melhor esforço.

O conceito de QoS (*Quality of Service* – Qualidade de Serviço) em redes de computadores é importante para que os serviços disponibilizados possuam algum nível de garantia e possam ser adequadamente caracterizados e conhecidos pelos usuários. Cada serviço utilizado que possui algum nível de garantia é descrito por um conjunto de parâmetros e o funcionamento de cada serviço deve ser consistente com sua descrição. Por exemplo, um serviço com atraso máximo de 15ms pode entregar informações com uma demora de 5ms, mas nunca com uma demora de 16ms. Garantir um comportamento consistente requer a gerência dos elementos envolvidos no fornecimento QoS. A inclusão de um novo roteador, por exemplo, não deve fazer com que um serviço com atraso de 15ms passe a operar com um atraso de 16ms. Assim, redes de computadores com QoS precisam ser gerenciadas, levando-se em conta não apenas as características amplamente conhecidas do paradigma de melhor esforço, mas também as características particulares de QoS.

1.1 Fornecimento de QoS

Fornecer QoS envolve a utilização de *protocolos de QoS.* As entidades de rede que suportam os protocolos, juntamente com os próprios protocolos de QoS, formam uma *arquitetura de QoS*. A arquitetura de QoS implantada fornece a seus usuários *serviços de QoS*. Em uma comunicação entre dois usuários de rede - sendo o usuário uma aplicação ou qualquer outra entidade que gere fluxos - o uso de serviços com QoS pode ocorrer em níveis e etapas diferentes.

Na rede de computadores, pode-se ter o fornecimento de QoS ao longo de todo o caminho da comunicação, apenas entre os equipamentos intermediários ou apenas em segmentos determinados. Nos sistemas finais (normalmente microcomputadores de usuários humanos), parâmetros de QoS podem ser verificados no protocolo de transporte, no sistema operacional e na aplicação geradora de um fluxo. Para que

garantias totais fim-a-fim possam ser alcançadas, serviços com QoS devem estar presentes em todos os níveis da comunicação. Por exemplo, se a rede de computadores garante uma entrega de quadros de vídeo a uma taxa constante, mas o sistema operacional do receptor não garante que o processo de apresentação seja escalonado também a uma taxa constante, então o vídeo poderá apresentar problemas de exibição.

O fornecimento de QoS em redes de computadores é alcançado quando protocolos de QoS são implantados nas estruturas da rede. Os protocolos de QoS podem operar em diferentes camadas do modelo de referência OSI [TAN 96], e podem fornecer diversos níveis de garantias. Os serviços de QoS que possuem garantias estritas utilizam-se de protocolos complexos; serviços de QoS com garantias menos estritas são implementados por protocolos menos complexos. Protocolos de QoS mais complexos exigem mudanças mais profundas nas estruturas das redes, enquanto que protocolos mais simples requerem mudanças mais superficiais.

A importância no fornecimento de QoS pode ser observada nas redes ATM [DUT 95], que foram definidas já com suporte a QoS desde seu início. Em redes IP, a importância do fornecimento de QoS pode ser verificada nos trabalhos do IETF [IET 2001] onde atualmente os protocolos de QoS mais importantes estão relacionados com as arquiteturas DiffServ (*Differentiated Services* - serviços diferenciados) [BLA 98], IntServ (*Integrated Services* - serviços integrados) [SHE 97], MPLS (*MultiProtocol Label Switching*) [ROS 2001] e SBM (*Subnet Bandwidth Management*) [YAV 99]. Tais arquiteturas não são definidas isoladamente: na verdade elas completam umas às outras para fornecer soluções de QoS mais abrangentes. Assim, os protocolos de cada arquitetura podem ser combinados para formar arquiteturas híbridas mais complexas.

A implantação de serviços com QoS é feita tipicamente em dois ambientes distintos: dentro dos domínios administrativos e entre os domínios administrativos. As arquiteturas de QoS podem ser implantadas para fornecer serviços de QoS em qualquer um dos dois ambientes. Entretanto, garantias estritas fim-a-fim só podem ser alcançadas quando as arquiteturas são implantadas nos dois ambientes ao mesmo tempo. Determinados protocolos são mais apropriados para garantias entre domínios (DiffServ, por exemplo), enquanto que outros são mais apropriados para garantir serviços dentro de um domínio (MPLS, por exemplo). Tem-se ainda protocolos que, em situações controladas, podem operar adequadamente nos dois ambientes (RSVP [WRO 97], por exemplo) mas que por esta razão são mais complexos.

Cada arquitetura de QoS possui suas características próprias, e implementa os serviços de fornecimento de QoS com mecanismos particulares. Como conseqüência, cada arquitetura também deve ser gerenciada de forma diferenciada e assim, a gerência de QoS fim-a-fim pode ser muito diversificada de acordo com a arquitetura de QoS utilizada.

1.2 Gerência de QoS

O termo "gerência de QoS" pode ser interpretado de duas formas distintas, conforme encontrado na literatura. A primeira forma refere-se aos mecanismos implantados nos elementos de rede necessários para que os níveis de QoS esperados possam ser alcançados. Por exemplo, os mecanismos para conformação de tráfego e policiamento em roteadores de borda de um domínio são utilizados para se gerenciar o

QoS de fluxos e agregados de dados.

Os mecanismos implantados na rede fornecem uma gerência automatizada, onde a intervenção humana não é observada *durante* o funcionamento de tais mecanismos. Percebe-se, entretanto, que intervenções fazem-se necessárias em pelo menos dois momentos: na implantação dos mecanismos de gerência, e na manutenção destes mecanismos.

A segunda forma de interpretação do termo "gerência de QoS" tem a ver com as facilidades disponibilizadas aos administradores de rede de forma que os mesmos possam ser capazes de manter as estruturas de QoS existentes em cada ambiente. Por exemplo, uma base de informações de gerência (MIB [MCC 90]) para DiffServ pode ser utilizada por um administrador para configurar a conformação de tráfego em um roteador de borda em um domínio administrativo.

As facilidades de gerência podem ser apresentadas de diversas formas aos administradores. Neste trabalho, define-se a expressão "arquitetura para gerência de QoS" como sendo uma arquiteturas formada pelo conjunto de facilidades de gerência de QoS disponibilizadas aos administradores de rede.

A principal diferença que existe entre as duas interpretações para "gerência de QoS" relaciona-se com a intervenção do administrador de redes humano. Na primeira interpretação o administrador não tem papel ativo no processo de gerenciamento: os mecanismos de fornecimento de QoS são os responsáveis por desempenharem a gerência. Na segunda interpretação, uma arquitetura de gerência fornece facilidades ao administrador, agora o responsável por desempenhar a gerência.

Alguns autores utilizam também o termo "controle de QoS" em conjunto com a primeira interpretação de "gerência de QoS" [MCD 99], onde a única diferença reside em questões temporais de controle. Neste trabalho, o controle de QoS será utilizado em substituição à primeira interpretação, enquanto que a gerência de QoS estará sempre relacionada com a segunda interpretação da expressão.

Como dito anteriormente, a gerência de QoS é utilizada na interação com as estruturas de fornecimento de QoS. Como conseqüência, a gerência de QoS envolve a gerência do controle de QoS nas estruturas de rede (sem a distinção dos termos anteriormente apresentados poderia-se dizer que existe a necessidade de uma "gerência de gerência de QoS", o que não é elegante). Concluindo, uma arquitetura de gerência de QoS fornece facilidades aos administradores de rede para que os mesmos possam gerenciar os vários aspectos do fornecimento de QoS, inclusive o controle de QoS.

1.3 Estado da arte

A necessidade de serviços de rede com QoS não é nova e é motivo de pesquisas que se desenvolvem já há bastante tempo [HUT 94]. As primeiras arquiteturas de rede foram elaboradas levando pouco ou nada em conta o conceito de QoS, o que não é adequado para as necessidades atuais. Arquiteturas mais recentes, como o ATM [DUT 95], foram criadas com serviços de QoS explicitamente definidos.

O interesse da comunidade científica pelo tema QoS foi reforçado à medida que a arquitetura TCP/IP [COM 91] passou a ser utilizada também como meio de transporte de informações com aspectos temporais críticos. As aplicações mais importantes da

Web (navegação interativa, transferência de arquivo e correio eletrônico) comportam-se bem em redes TCP/IP, porque as necessidades de QoS são pouco estritas, quando existem. Entretanto, aplicações mais especializadas, que passaram a usar o TCP/IP, não operam corretamente. Para tais aplicações, a existência de QoS em redes TCP/IP é imprescindível.

Várias pesquisas começaram a ser desenvolvidas para tentar resolver o problema. Algumas pesquisas acadêmicas mereceram destaque e um trabalho anterior deste autor, na forma de exame de qualificação, analisa tais pesquisas [GRA 2000]. O fornecimento de QoS em redes TCP/IP começou a se tornar mais efetivo a partir do momento em que o IETF [IET 2001] passou a criar grupos de trabalhos voltados a esta área. Como vários integrantes destes grupos pertencem à indústria de equipamentos de rede, as definições do IETF passam rapidamente de padrões de direito para padrões de fato.

Atualmente muitas pesquisas estão em desenvolvimento. As principais arquiteturas de fornecimento de QoS, que possuem grande potencial de utilização em ambientes reais, são desenvolvidas dentro do IETF. A arquitetura DiffServ procura fornecer QoS através da priorização e tratamento especial de fluxos agregados, enquanto que o protocolo MPLS identifica rotas inteiras que devem ser utilizadas. A arquitetura IntServ utiliza um protocolo de sinalização (o RSVP) capaz de reservar recursos ao longo do caminho de um fluxo. Por fim, as definições do SBM procuram garantir QoS já no nível 2 através de priorização de quadros IEEE 802. Além dessas soluções, existem ainda as soluções das redes onde o conceito de QoS foi incorporado desde as primeiras definições. Aqui encontram-se, por exemplo, as redes Frame Relay [STA 98] e ATM [DUT 95].

A gerência de QoS é uma área nova, já com pesquisas em andamento, mas muitas soluções existentes são baseada na utilização de estruturas de gerência anteriormente aplicadas ao gerenciamento de serviços de melhor esforço. Isso pode ser especialmente notado quando se verificam os trabalhos do IETF: todas as arquiteturas de QoS possuem documentos (RFCs ou *drafts*) para a definição de MIBs SNMP [STA 99] no suporte aos serviços.

Uma alternativa relativamente recente e de grande sucesso no meio acadêmico e comercial é a utilização do gerenciamento baseado em políticas (PBNM – *Policy-Based Network Management*) aplicado à gerência de QoS [QOS 99]. O PBNM permite a abstração de complexidades de rede sem que com isso perca-se o controle sobre as estruturas. O administrador de rede define, em uma linguagem de alto nível, políticas que devem ser implantadas. Na gerência de QoS o administrador passa a definir políticas que tratam aspectos de QoS. Quando uma política definida é implantada na rede, componentes especiais interpretam as definições feitas pelo administrador e traduzem tais definições para ações na rede gerenciada. Como políticas idênticas podem ser implantadas em arquiteturas de QoS diferentes, cada componente traduz as políticas para ações distintas em cada arquitetura.

Entretanto, a utilização de PBNM não é capaz de fornecer todas as facilidades de gerência que um administrador de redes com QoS necessita. Vários outros aspectos não cobertos pelas políticas também precisam ser gerenciados, aspectos estes verificados ao longo da tese.

1.4 Motivação

A introdução de uma arquitetura de QoS melhora os serviços disponibilizados aos usuários de redes, mas também aumenta a complexidade da gerência dessa rede [GRA 2000a]. Além disso, uma rede com QoS só poderá fornecer serviços com garantias se a arquitetura de QoS utilizada for adequadamente gerenciada. As arquiteturas de fornecimento de QoS não garantem, por si só, o correto funcionamento dos serviços disponibilizados. Sem uma solução de gerência que permita aos administradores de redes atuar sobre as arquiteturas, o fornecimento adequado de QoS está comprometido. Assim, o principal argumento desta tese é de que para se ter garantias de funcionamento dos serviços de redes com QoS são necessários:

- Uma arquitetura para fornecimento de QoS;
- Um sistema de gerência de QoS.

Como dito na seção anterior, a gerência baseada em políticas é importante, mas não é suficiente do ponto de vista dos administradores da rede. Um sistema de gerência de QoS deve suportar políticas, mas também deve suportar outras funcionalidades necessárias à manutenção dos serviços com QoS.

Analisando o processo de gerência de QoS em uma rede de computadores, pode-se verificar que diversas etapas da gerência exigem facilidades auxiliares distintas. A implantação de uma rede com QoS é melhor realizada se uma ferramenta de pré-analise indicar problemas não percebidos pelo administrador de rede. Em redes já implantadas, uma ferramenta para descoberta de estruturas de QoS na rede pode auxiliar o gerente na utilização de características presentes anteriormente, mas por falta de informação, ainda não utilizadas. Quando todas as estruturas de QoS estiverem operacionais, a gerência baseada em políticas ajuda o administrador a definir os comportamentos a serem aplicados no ambiente gerenciado. A monitoração de QoS indica quais os pontos problemáticos de rede que apresentam degradações.

Para que a gerência de QoS possa ser efetiva, o ambiente de gerência deve ser capaz de explicitamente fornecer dados sobre os serviços de QoS implantados. O fornecimento explícito deve ser tal que o administrador da rede não perca muito tempo procurando informações de QoS no meio de todas as outras informações relativas à gerência dos outros aspectos da rede.

A motivação principal para a realização deste trabalho é a definição de um modelo integrado de gerência de rede onde os aspectos de QoS sejam levados em conta de forma explícita. Um ambiente que segue o modelo proposto deve ser capaz de auxiliar o administrador da rede na implantação e manutenção de diferentes arquiteturas de QoS através de ferramentas integradas em um mesmo ambiente de gerência.

Para que a definição de tal modelo seja possível, um conjunto de estudos foi realizado de forma a identificar os aspectos de gerência de QoS importantes aos administradores de rede. A partir destes aspectos, facilidades foram definidas e integradas, formando o modelo.

1.5 Objetivos, contribuições e organização da tese

Os principais objetivos da tese são:

- Classificação das soluções para gerência de QoS em tarefas de gerenciamento de QoS;
- Criação de um modelo genérico que permita a integração das diversas facilidades auxiliares necessárias à gerência de uma rede heterogênea que possua uma arquitetura de QoS implantada (possivelmente com protocolos e estruturas de QoS diferentes operando ao mesmo tempo na mesma rede);
- Analisar o modelo criado em relação às suas funcionalidades e características.
- Aplicar o modelo através da definição de uma arquitetura de gerência de QoS para redes IP.

A tese apresenta como resultado final as seguintes contribuições:

- A determinação de quais as etapas e situações em que a gerência de QoS se faz necessária, permitindo ao administrador da rede uma sistematização na utilização das facilidades de gerência de QoS;
- A definição de um modelo geral que permite a gerência integrada de QoS nas etapas e situações determinadas como mais importantes na tese;
- Um ambiente baseado na Web para gerência integrada e explícita de redes com arquiteturas de QoS em redes IP.

Esta tese está organizada em 6 capítulos. O capítulo 2 discute sobre gerência de QoS e o uso de gerenciamento de redes baseado em políticas. Ao final, o problema investigado nesta tese de doutorado é definido. O capítulo 3 apresenta a classificação das tarefas de gerência de QoS realizada de forma a identificar quais os principais aspectos de QoS a serem suportados por soluções de gerência de QoS. No capítulo 4 é proposto o modelo para gerência integrada de QoS em redes de computadores. O modelo é definido com base nas tarefas de gerência do capítulo 3. O modelo proposto é analisado no capítulo 5, onde questões como exeqüibilidade e escalabilidade são discutidas. Por fim, o capítulo 6 encerra esta tese apresentando as conclusões do trabalho.

2 Gerência de QoS e definição do problema

A gerência de rede tradicional lida com um conjunto de informações de rede que cresce constantemente, tanto em diversidade quanto em volume. Os administradores de rede tratam com informações de fontes diferenciadas, e como redes de computadores são tipicamente heterogêneas, isso acaba tornando a gerência mais complexa. Além disso, o volume de informações é proporcional ao tamanho da rede, e como a disseminação da Internet incentiva a interconectividade dos usuários, as redes tendem a se tornar cada vez maiores, e logo, com mais informações a serem gerenciadas.

Não bastasse o conjunto de dados de gerência gerados pela diversidade e pelo número de dispositivos de uma rede, a existência de uma arquitetura de fornecimento de QoS inclui um conjunto a mais de dados que torna o universo total de informações de gerência extremamente grande e complexo, e que não pode mais ser tratado pelas soluções de gerência de redes tradicionais [EDE 2001].

Neste capítulo serão revistos o modelo de gerência tradicional orientado a dispositivos, a gerência de redes baseada em políticas, e uma revisão sobre as soluções de gerência de QoS na seção de trabalhos relacionados. Por fim, o capítulo encerra com definindo o problema investigado nesta tese: a falta de interoperabilidade entre soluções de gerência de QoS.

2.1 Gerência orientada a dispositivos

As plataformas de gerência atuais são orientadas a dispositivos, no sentido de que os administradores investigam cada equipamento de rede particular cadastrado no sistema de gerência à procura de informações. A granularidade da gerência é extremamente pequena, já que o gerente pode ter acesso a informações específicas sobre qualquer dispositivo cadastrado. A FIGURA 2.1 apresenta a interface gráfica da plataforma de gerência HP OpenView [HEW 2001] orientada a dispositivos.

O uso de mapas de rede facilita o processo de investigação porque a hierarquização dos dispositivos normalmente segue a estrutura de conectividade da rede gerenciada. É intuitivo a um administrador de rede navegar por mapas que reflitam o ambiente que se está gerenciando. Além disso, o uso de mapas permite a abstração de segmentos menos importantes, que são confinados em nuvens (abstração visual normalmente utilizada para representar sub-redes) que podem ser visitadas apenas em situações críticas ou especiais.

Entretanto, na tentativa de colher informações sobre o QoS da rede gerenciada, o administrador da rede acaba sendo obrigado a investigar cada dispositivo em particular, para verificar se os mesmos possuem informações e serviços oriundos da arquitetura de fornecimento de QoS. Para redes pequenas, este tipo de gerência pode ser utilizado, mas em redes maiores seria impossível a um administrador analisar todas as informações existentes.

Neste contexto, a gerência orientada a dispositivos, ainda que preciosa pelos motivos citados anteriormente, não fornece facilidades adequadas de gerência de QoS.

São necessários processos onde a granularidade do acesso aos dispositivos seja maior que a encontrada na gerência padrão, de forma que o administrador de rede tenha uma visão mais global da rede, sem que com isso se perca o controle sobre a mesma.

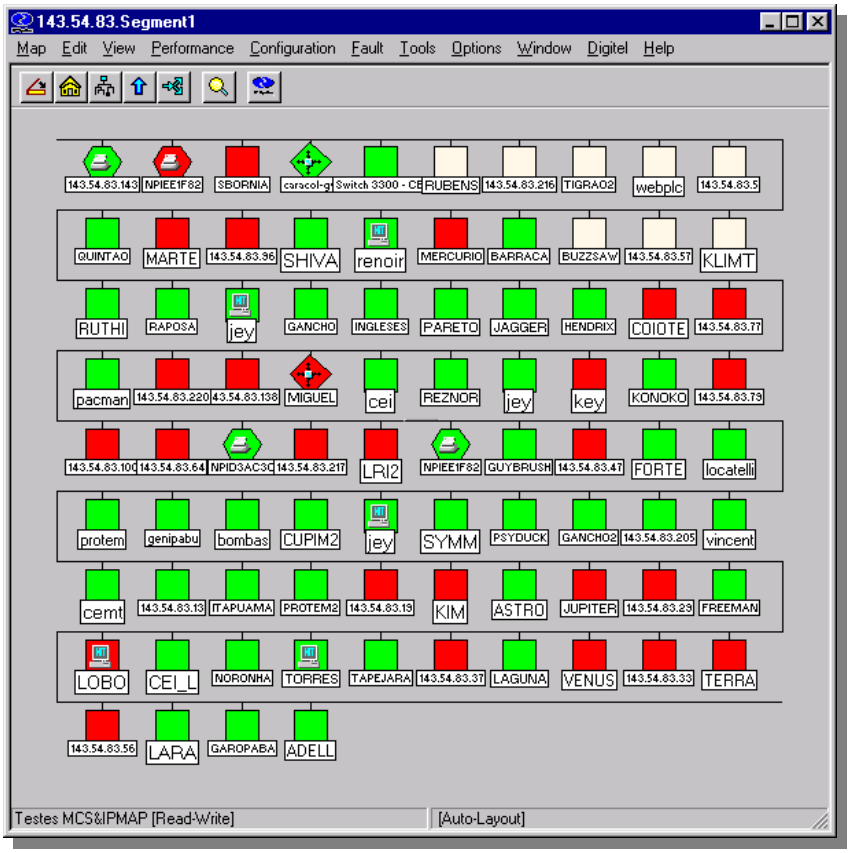

FIGURA 2.1 – Exemplo de uma solução de gerência orientada a dispositivos

Logo, é uma necessidade real a existência de mecanismos capazes de abstrair o grande conjunto de informações disponibilizadas, e capazes de automatizar as tarefas de gerenciamento, complementado a gerência orientada a dispositivos das plataformas atuais através de soluções que forneçam uma visão abrangente da rede e da arquitetura de fornecimento de QoS existente.

A seção a seguir apresenta o gerenciamento de redes baseado em políticas, que atualmente representa a solução mais promissora para as questões levantadas até este momento.

2.2 Gerenciamento de redes baseado em políticas (PBNM)

O PBNM procura introduzir um nível de abstração de informações de gerenciamento maior para facilitar as tarefas de gerência a serem executadas. Quando o administrador da rede passa a lidar com muitas estruturas diferentes, a gerência passa a ser complexa, e conseqüentemente mais difícil de ser mantida.

O uso de PBNM pode ser comparado ao uso das linguagens de programação. Neste caso, cada sistema possui um conjunto de primitivas de baixo nível que permite a programação do mesmo. As primitivas são codificas em linguagem *assembler*, que

apesar de garantir o controle total sobre a máquina, é muito complexa. Cada conjunto de primitivas é dependente de plataforma. Por outro lado, as sintaxes das linguagens de programação de mais alto nível são independentes de plataforma e mais simples que a linguagem de máquina. Os vários padrões de gerência podem ser comparados às linguagens de máquina, enquanto que o PBNM é comparado com as linguagens de programação de alto nível, independentes de plataforma.

Com a abstração fornecida, o administrador de rede preocupa-se em determinar as políticas de gerência a serem usadas. As tecnologias utilizadas preocupam-se em interpretar estas políticas e implantá-las na rede. O fornecimento de QoS em redes heterogêneas é rico em diversidade de tecnologias. A gerência de todas as tecnologias necessárias é complexa, e o uso de políticas para a gerência de QoS, neste contexto, é uma solução interessante.

Uma política é, em essência, uma ou mais regras que descrevem ações que devem ocorrer quando condições específicas existirem na rede. Uma regra pode ser formada por uma combinação de outras regras. Como conseqüência, uma política pode ser formada pela combinação de outras políticas. A hierarquia de políticas é essencial para o sistema de gerência, porque permite que políticas complexas possam ser formadas pela combinação de várias políticas simples.

Como definido pelo grupo de trabalho policy (*Policy Framework*) [HAL 2000] do IETF [IET 2001], uma arquitetura para gerência de QoS baseada em políticas [QOS 99] define pontos de ação (PEP – *Policy Enforcement Point*) e pontos de decisão (PDP – *Policy Decision Point*) na rede. Os PEPs aplicam as políticas definidas, enquanto que os PDPs tomam decisões baseados em políticas recuperadas de um repositório de políticas (FIGURA 2.2).

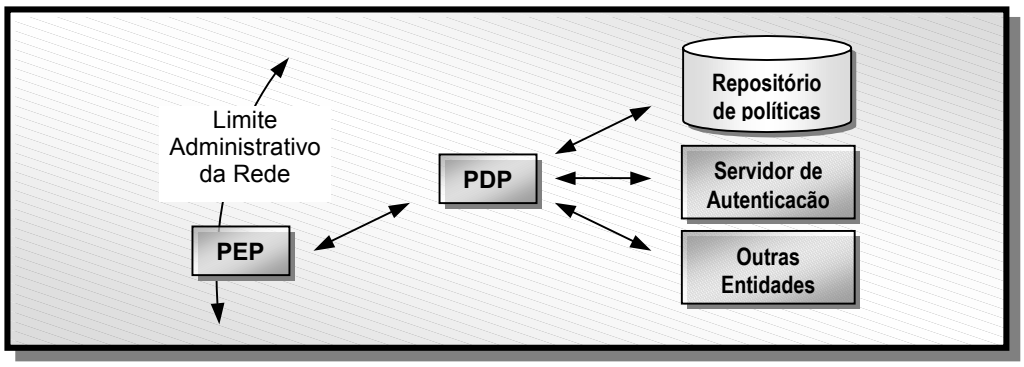

FIGURA 2.2 – Arquitetura para gerenciamento de QoS baseado em políticas

A existência de vários PEPs é interessante porque os pontos de ação são colocados em vários locais possíveis. Tipicamente os pontos de ação estão localizados nos limites administrativos da rede. Na FIGURA 2.2, poderiam existir vários PEPs e alguns PDPs. A existência de mais de um PDP aumenta a complexidade da arquitetura porque as decisões passam a ser distribuídas. Por outro lado, tem-se um aumento na robustez da solução, porque se um PDP deixar de funcionar os outros podem assumir seu lugar.

A gerência de QoS se dá através da interação entre os PDPs e o ambiente de gerência da rede. Uma estação de gerência deve ser capaz de determinar novas políticas, seu momento de utilização e em que pontos ela deve ser aplicada. Assim, uma

comunicação entre os PDPs e uma estação de gerência deve ser fornecida.

As políticas definidas são armazenadas em um repositório de políticas. Dentro de cada PDP existe uma base de dados de políticas (PIB – *Policy Information Base*) a serem consideradas. A representação das políticas na PIB é própria e define, através de classes hierárquicas, os vários parâmetros de cada política a ser aplicada pelos PDPs. A programação da PIB gera, indiretamente, uma atualização do repositório de políticas.

A estação de gerência acessa a PIB de cada PDP através de SNMP. A visão do gerente é obtida através do mapeamento de PIB para uma MIB específica de gerência de políticas. Os dados da PIB são disponibilizados na MIB e esta acessada pelo gerente via SNMP (FIGURA 2.3).

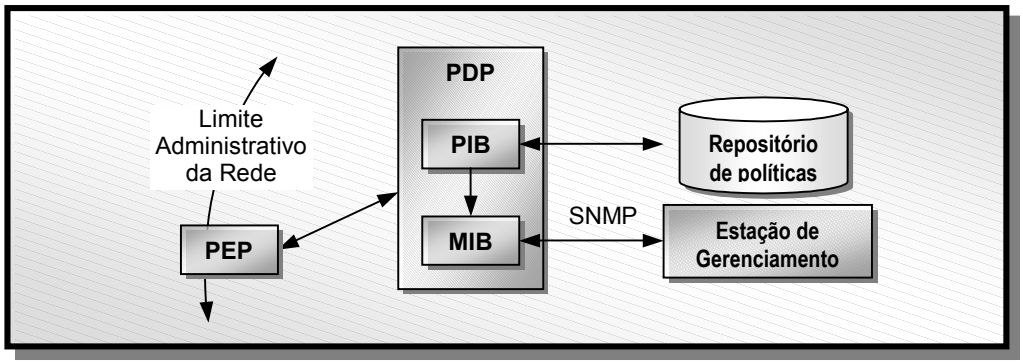

FIGURA 2.3 – PDP, PEP e estação de gerenciamento

Uma característica importante da gerência de QoS baseada em políticas é que ela pode ser aplicada em conjunto com a gerência padrão. Em uma mesma rede, pode-se ter o gerenciamento das estruturas para fornecimento de QoS através da gerência baseada em política e também através do gerenciamento padrão. Por exemplo, uma política pode determinar que um fluxo de vídeo que cruza um limite administrativo deve ter um *jitter* nulo. Isso pode ser implementado através de DiffServ. Os roteadores de borda são programados pelo PEP associado para que tratem adequadamente o fluxo de vídeo. Esta programação pode ser realizada através da comunicação, via SNMP, do PEP com o roteador. A MIB DiffServ do roteador é acessada para tal. O mesmo acontece com o RSVP [WRO 97]. O protocolo pode ser utilizado por um PEP para estabelecer um tratamento de um fluxo interno entre dois roteadores de alta velocidade.

A coexistência entre arquitetura de QoS e o PBNM é possível porque, como dito anteriormente, as arquiteturas são comparadas à linguagem de máquina, enquanto que as políticas são comparadas às linguagens de mais alto nível. No exemplo anterior, DiffServ e RSVP (IntServ) operam em um mesmo nível; a gerência de QoS baseada em políticas opera em um nível acima, coordenado às estruturas do nível inferior.

2.3 PBNM, pesquisas e o mercado de soluções de gerência

O PBNM vem sendo investigado já há algum tempo por grupos de pesquisa. A Imperial College London foi a pioneira na área através dos trabalhos desenvolvidos principalmente por Emil Lupu e Morris Sloman [MOF 93] [SLO 94] [LUP 99]. A importância acadêmica do PBNM acabou refletindo no modelo do IETF, apresentado anteriormente. Vários outros aspectos envolvendo políticas ainda estão sendo

investigados, e resultados parciais são encontrados em documentos IETF na forma de *draft*. Entre estes, o trabalho de definição de um sistema de gerência para políticas, descrito por H. Mahon et al. [MAH 2000], é de especial importância no contexto desta tese, pois é a primeira tentativa do IETF de padronização de um modelo não apenas para se implantar e controlar políticas, mas também para gerenciá-las, preocupando-se aqui com a interação com o administrador da rede. Este trabalho do IETF é uma das bases do modelo de gerenciamento de QoS proposto na tese e apresentado mais a frente.

Na indústria, as soluções para gerência de QoS comerciais ganharam um destaque recente também como conseqüência das propostas de PBNM. Os principais produtores de soluções de mercado para gerência de redes (principalmente para redes baseadas em IP) lançaram suas próprias soluções PBNM, muitas vezes integradas em plataformas de gerência tradicionais.

A Hewlett-Packard criou o PolicyXpert [HEW 2001a] que faz parte do conjunto de ferramentas de gerência do pacote OpenView. A Cisco também produziu sua solução PBNM e lançou, em 1999, o QPM (QoS Policy Manager) como parte de sua iniciativa de gerência de política mais ampla denominada CiscoAssure [CIS 2000]. A Extreme Networks fez o mesmo e criou o EPICenter (anteriormente conhecido como ExtremeAware Enterprize Manager - EEM) [EXT 2001]. Outros fornecedores também lançaram suas soluções PBNM, como por exemplo, a Nortel Networks, Lucent Techonologies e Orchestream Networks [SAU 99].

Apesar de estas soluções poderem apresentar um eventual sucesso mercadológico, todas sofrem da falta de integração por não serem totalmente baseadas nas definições do IETF. Por exemplo, apesar do IETF defender o uso de um serviço de diretório como o LDAP [STR 2001] para o armazenamento de políticas, todas as soluções de mercado acabam utilizando sistemas de banco de dados para este fim. A solução da Extreme Networks, por exemplo, utiliza Sybase [PAN 96], enquanto que o Orchestream Enterprise da Orchestream Networks é baseado em Oracle [LON 2000]. Como conseqüência, os usuários de tais sistemas inevitavelmente enfrentarão problemas quando a interconexão entre os sistemas de gerência de políticas passar a ser necessária [CLA 2000].

2.4 Análise sobre a abrangência do PBNM

É importante perceber que a solução PBNM não substitui as formas de gerência padrão, mas sim complementa estas formas permitindo ao administrador a definição de políticas globais. Por outro lado, o PBNM só é possível mediante a existência dos seguintes pré-requisitos:

- A rede gerenciada já deve possuir uma arquitetura de QoS implantada;
- A arquitetura implantada deve ser conhecida pelo PBNM;
- Os pontos de atuação e decisão de políticas (PEPs e PDPs) já devem estar definidos;

Além disso, observam-se ainda as seguintes características do PBNM:

- Uma política aplicada só é verificada no momento de sua implantação. Se a política não se comportar como o esperado, o PBNM não é responsável por sinalizar esta situação ao administrador;

- Novos equipamentos que implementam funcionalidades relativas ao fornecimento de QoS devem ser manualmente cadastrados no sistema. O PBNM não inclui nenhum mecanismo de descoberta automática.

Estas características indicam que o PBNM só pode ser aplicado depois que todas as estruturas de rede forem adequadamente configuradas para suportarem este tipo de gerência. Antes disso, o administrador da rede deve proceder com diversas tarefas relacionadas ao QoS que não são cobertas pelo PBNM. Além disso, existem ainda outras tarefas que precisam ser realizadas, mesmo depois do PBNM ter sido implantado, e que também não são suportadas no PBNM.

Assim, pode-se dizer que o PBNM é uma solução que abrange apenas alguns aspectos da gerência de QoS. Diversas outras atividades relacionadas à gerência de QoS devem ser realizadas, ainda que o PBNM esteja presente na rede gerenciada.

2.5 Trabalhos relacionados à gerência de QoS

Aspectos de gerência de QoS não cobertos pelo PBNM acabam sendo investigados em outras soluções pesquisadas. Nesta seção, alguns trabalhos relacionados a diversos aspectos da gerência de QoS são apresentados, com o objetivo de se traçar um panorama mais geral em relação a utilização destas soluções.

Hong. et al. [HON 99] propôs, em 1999, uma solução para gerência baseada em CORBA para o gerenciamento do QoS de serviços multimídia distribuídos e de aplicações que pertenciam à plataforma MAESTRO [YUN 97]. Na solução, a arquitetura em camadas do sistema de gerência permite uma gerência fim-a-fim dos mecanismos de fornecimento de QoS. O sistema possui, por exemplo, funcionalidades para especificação de QoS, mapeamento de QoS entre níveis diferentes da hierarquia de protocolos, controle de admissão e negociação e renegociação de parâmetros de QoS. Uma MIB QoS genérica foi desenvolvida no projeto para permitir o acesso aos parâmetros de QoS em tal arquitetura hierárquica.

Em 2000, o autor desta tese também propôs uma MIB QoS como meio de programação dos processos DiffServ de marcação de pacotes nos sistemas finais [GRA 2000b]. Na proposta, o SNMP também era utilizado para permitir aos usuários de uma rede a reserva de recursos de QoS através de solicitações de tais recursos a um negociador de banda (BB – *Bandwidth Broker* [NIC 99]).

Jiang et al. [JIA 2000] apresentou uma análise sobre estratégias para a monitoração de QoS em redes. O trabalho definia que duas informações principais precisavam ser fornecidas por um sistema de monitoração:

- A identificação de degradações fim-a-fim do QoS de fluxos de interesse; e
- Mediante degradações, a identificação de quais pontos da rede eram os responsáveis pelo mau desempenho.

A identificação do QoS fim-a-fim é facilmente realizada, hoje em dia, com a utilização de protocolos capazes de informar as condições da rede aos sistemas finais. Tipicamente, os protocolos RTP/RTCP [SCH 96] são utilizados para este fim. Entretanto, a identificação dos pontos de degradação da rede é uma tarefa mais complexa porque exige instrumentação dos dispositivos da rede de forma que estes possam devolver informações sobre os fluxos correntes. Joshi et al. apresentou, neste

sentido, um trabalho [JOS 2000], em 2000, que integrava a monitoração de QoS à gerência de redes padrão.

O IETF também define uma solução para inspecionar a rede e colher estatísticas sobre seu desempenho [BRO 97]. Uma implementação dessa proposta é encontrada no projeto NeTraMet [BRO 2001]. Um administrador de rede é responsável por definir fluxos de interesse e repassar estas definições a medidores de fluxos encontrados em PCs (com sistema operacional DOS ou Unix). Coletores de amostras entram então em contato com os medidores, via SNMP, para recolher informações sobre os fluxos observados [BRO 97a]. Neste instante, então, os fluxos podem ser analisados e o QoS fornecido pela rede computado. Entretanto, a derivação do QoS fornecido é de responsabilidade do usuário do NeTraMet.

Outra solução, neste caso não baseada nas definições do IETF, é o ntop [DER 2001]. Aqui, os fluxos também devem ser definidos e repassados a um monitor capaz de verificar o tráfego da rede. Novamente, a derivação de problemas de degradação de QoS são de responsabilidade do usuário do sistema, e não do sistema em si.

Outro aspecto importante em relação à gerência de QoS está relacionado com a identificação de dispositivos e caminhos em uma rede capazes de fornecer níveis de serviços mais adequados. Neste sentido, o IETF produziu a RFC 2990 [HUS 2000] que, entre vários de seus itens, aponta para a necessidade da existência de facilidades para a descoberta de caminhos alternativos à rota padrão de melhor esforço entre máquinas de uma rede. Tanto os serviços integrados quantos os serviços diferenciados utilizam como base a rota padrão para a diferenciação dos fluxos entre dois sistemas finais. Entretanto, outras rotas alternativas poderiam ser utilizadas para garantir que o QoS contratado seja realmente cumprido.

2.6 Definição do problema

As propostas e soluções de gerência de QoS apresentadas anteriormente são obviamente importantes. Entretanto, cada solução é definida separadamente, no máximo interagindo com soluções de gerência padrão, como acontece com as soluções PBNM de mercado. Apesar do IETF possuir definições para vários aspectos de QoS (por exemplo, PBNM [YAV 2000] e monitoração para medição de desempenho de rede [BRO 97]), nem sempre as soluções desenvolvidas seguem tais definições. Como resultado final, as soluções que tratam aspectos diferentes do QoS não possuem integração entre si. Neste cenário, os administradores de rede acabam na prática sendo forçados a utilizar diversas ferramentas para gerenciar os diversos aspectos de QoS existentes.

Acredita-se que a falta de integração é também resultado da não existência de uma classificação clara sobre o processo de gerência de QoS. Na seção anterior sobre os trabalhos relacionados, pôde-se constatar a existência de soluções, por exemplo, para a coordenação de QoS fim-a-fim, monitoração de QoS e descoberta de rotas alternativas. Estas atividades podem ser mapeadas para as cinco áreas funcionais de gerência definidas pela OSI (gerência de falhas, configuração, contabilização, desempenho e segurança) [LEI 96]. Entretanto, acredita-se que uma classificação específica para gerência de QoS poderia contribuir para a integração entre as ferramentas existentes.

No próximo capítulo, uma classificação específica para gerência de QoS é apresentada. Com tal classificação, as soluções podem ser definidas em tarefas de

gerência de QoS (como será apresentado). Tais tarefas já podem ser hoje realizadas utilizando ferramentas de mercado. Novamente, a integração entre tais ferramentas não existe. Com o intuito de promover tal integração, é ainda definido um modelo para gerência integrada de QoS capaz de dar suporte às tarefas de gerência de QoS a serem definidas.

Resumidamente, o problema abordado nesta tese de doutorado é a falta de classificação das soluções de gerência de QoS, e a falta de integração entre os sistemas existentes. A falta de uma classificação levou à definição de tarefas de gerência de QoS, e a falta de integração levou à proposta de um modelo para gerenciamento integrado capaz de fornecer facilidades para a execução das tarefas definidas.

3 Tarefas de gerência de QoS

Como visto no capítulo 2, a gerência padrão de redes é orientada a dispositivos, onde o gerente deve investigar cada dispositivo individualmente à procura de informações importantes ao gerenciamento. O PBNM tenta resolver este problema, mas como constatado, a solução é restrita a alguns aspectos da gerência de QoS. Vários outros aspectos acabam sendo gerenciados por soluções diversas e não integradas, levando a uma situação onde a integração se torna necessária.

Para uma gerência de QoS mais efetiva, este capítulo apresenta a primeira contribuição da tese: a classificação da gerência de QoS em "tarefas relacionadas ao gerenciamento de QoS" [GRA 2001] [GRA 2001f]. Tais tarefas visam cobrir grande parte das atividades relacionadas à gerência de QoS como, por exemplo, àquelas relacionadas às soluções de gerência de QoS apresentadas no capítulo anterior.

A classificação é resultado de pesquisas realizadas em relação aos trabalhos desenvolvidos em outras universidades e em alguns produtos comerciais. O conjunto das tarefas de gerência de QoS é composto por: implantação, descoberta, manutenção, monitoração, análise e visualização de QoS (apresentadas nas próximas seções). Primeiras versões da classificação deram origem ao modelo de gerenciamento integrado apresentado na capítulo 4. A existência do modelo permitiu realizar reavaliações e ajustes na classificação, onde novas tarefas de gerência foram então definidas (por exemplo, análise de QoS), outras tiveram seu escopo melhor delimitado (descoberta de QoS) e terceiras tiveram seu escopo reduzidos (monitoração de QoS). Algumas tarefas foram fruto direto das pesquisas (manutenção e monitoração de QoS, por exemplo), enquanto outras foram intuitivamente criadas mas posteriormente validadas (por exemplo, descoberta de QoS). Por fim, algumas tarefas não possuem ainda pesquisas efetivas desenvolvidas por outros grupos, mas devem ser realizadas em estágios futuros da gerência de QoS (por exemplo, análise e visualização de QoS).

As seções a seguir apresentam cada uma das tarefas de gerência de QoS definidas na classificação, e eventualmente algumas etapas da criação destas tarefas serão novamente discutidas. Ao final do capítulo as últimas sessões apresentarão o relacionamento entre as tarefas, e exemplificarão como estas interagem umas com as outras. Isso acaba por reforçar ainda mais a necessidade de uma integração do gerenciamento de QoS.

3.1 Implantação de QoS

Nesta seção, a nomenclatura utilizada pelo QoS fórum [QOS 2001] é utilizada para a definição da implantação de QoS. Segundo esta nomenclatura, são definidos protocolos e arquiteturas de QoS. Os protocolos de QoS são efetivamente as soluções particulares que implementam QoS em uma rede. Serviços diferenciados, RSVP, MPLS e SBM são exemplo, do ponto de vista do QoS fórum, de protocolos de QoS. Uma arquitetura de QoS é o resultado da utilização, em uma mesma rede, de vários protocolos. Assim, as composições entre, por exemplo, serviços diferenciados e RSVP definem uma arquitetura. Serviços integrados e MPLS formam outra arquitetura.

Atualmente, as soluções e protocolos de QoS mais destacados são aqueles desenvolvidas pelo IETF, principalmente MPLS [ROS 2001], DiffServ [BLA 98] e IntServ [SHE 97]. Cada solução, na verdade, pode ser disponibilizada usando diferentes configurações. Por exemplo, uma solução pode usar negociadores de banda (BB – *Bandwidth Brokers*) [NIC 99] que permitem a alocação dinâmica de recursos em serviços diferenciados. Outra solução pode implantar DiffServ sem negociadores instalados na rede [GRA 2000c].

Assim, pode-se esperar uma variedade de arquiteturas de QoS tão ampla quanto o número de possíveis combinações entre protocolos de QoS e suas variações. Como conseqüência, a escolha de uma determinada arquitetura de QoS acaba se tornando, por parte do administrador da rede, uma tarefa sensivelmente complexa já que cada rede gerenciada acaba implantando uma arquitetura de QoS diferente pois os requisitos e objetivos das redes são diferentes. Neste contexto, a implantação de QoS é a tarefa de se escolher e implantar adequadamente soluções de QoS compatíveis com a rede gerenciada. O resultado da implantação de QoS é uma arquitetura de QoS única que fornece serviços a uma rede particular. A arquitetura resultante é uma coleção de soluções e protocolos de QoS que juntos oferecem serviços aos usuários da rede de computadores.

A implantação de QoS aponta para a necessidade de *softwares* que aconselhem os administradores de rede sobre as possíveis arquiteturas de QoS, suas configurações e problemas. Uma arquitetura de QoS pode ser mais adequada para uma rede em particular, mas para outra rede a mesma arquitetura pode ser totalmente inapropriada.

A arquitetura final derivada por um possível *software* de auxílio à implantação de QoS deve ser o resultado da análise dos seguintes elementos:

- **Topologia de rede**. Cada rede possui uma topologia particular e para cada topologia uma arquitetura de QoS particular é mais adequada;
- **Tráfego de rede**. Mesmo com topologias idênticas, cada rede apresenta tráfegos diferentes. Uma rede pode ter mais tráfego de saída que de entrada (por que seus usuários acessam conteúdo multimídia externo, por exemplo), mais tráfego interno que externo (porque os usuários acessam mais o servidor Web da Intranet e menos servidores Web da Internet, por exemplo);
- **Prioridades das aplicações de rede**. O tráfego da rede é principalmente gerado por aplicações (a rede em si gera pouco tráfego). A prioridade de cada aplicação varia de rede para rede. Por exemplo, o tráfego gerado pela manutenção da base de dados de uma empresa nos finais de tarde pode ter maior prioridade que o tráfego resultante da navegação Web de seus funcionários;
- **Suporte aos protocolos de QoS nos dispositivos de rede**. A arquitetura final só pode ser aplicada se os protocolos de QoS selecionados forem suportadas pelos dispositivos de rede. Caso contrário, a arquitetura final deve ser novamente computada levando-se então em conta o suporte existente nos dispositivos aos protocolos de QoS.

A determinação da arquitetura final de QoS pode ser complementada com a utilização de soluções de simulação de rede. Neste caso, espera-se que o *software* de implantação forneça facilidades para que os administradores descrevam graficamente a

topologia da rede, o tipo de tráfego associado e as prioridades das aplicações. As simulações podem então ser executadas para que o administrador inicie a verificação do comportamento de cada possível protocolo de QoS. Como visto, protocolos diferentes podem existir ao mesmo tempo no mesmo ambiente, de forma a colaborarem entre si, formando uma arquitetura de QoS. Assim, o *software* de implantação também deve fornecer funcionalidades para que seja possível a um administrador de rede, por exemplo, testar a utilização de mecanismos de reserva de recursos como o RSVP [BRA 97] internamente ao domínio administrativo, e usar DiffServ nos limites da rede.

A escolha de uma arquitetura de QoS é uma tarefa que pode ser realizada sem a efetiva interação com a rede gerenciada. A escolha é feita *offline*, mas posteriormente deve ser implantada para que obviamente apresente algum efeito. Após decidir sobre a arquitetura final a ser utilizada, os administradores iniciam a configuração dos dispositivos, trocando fisicamente aqueles que são inapropriados, atualizando o sistema operacional de outros, etc. Os procedimentos para a implantação de uma arquitetura podem ser então realizados de duas formas distintas: localmente ou remotamente. Procedimentos realizados localmente consomem muito tempo porque exigem o deslocamento físico do administrador até o dispositivo de interesse. Assim, deve ser priorizada a utilização de ferramentas de configuração que sejam capazes de manter o número de intervenções locais pequeno.

A configuração e atualização de *software* dos dispositivos (por exemplo, a configuração de uma disciplina de filas ou a atualização da versão do sistema operacional de um roteador) podem ser feitas remotamente em quase todos os casos, e realizada principalmente através de sessões Telnet, acesso HTTP em dispositivos mais modernos, e usando interações agente/gerente SNMP [CAS 90] [STA 99]. A alteração física de equipamentos é essencialmente local e, obviamente, não pode ser realizada remotamente.

Resumidamente, na implantação de QoS há a execução das seguintes etapas:

1) Levantamento da topologia física e lógica da rede;
2) Determinação das prioridades das aplicações;
3) Simulação das arquiteturas de fornecimento de QoS na rede existente e suas aplicações;
4) Configuração e/ou alteração de equipamentos para que a rede suporte a arquitetura de QoS escolhida.

A implantação de QoS pode ser realiza com o auxílio de ferramentas já disponíveis. Isso facilita nas atividades de definição da rede, simulação das arquiteturas e sua efetiva implantação.

As soluções para topologia de rede visam auxiliar os gerentes na definição gráfica da rede gerenciada. Tal definição deve conter os dispositivos que apresentam serviços com QoS de forma a permitir a observação da arquitetura de QoS implantada. Um exemplo de solução de topologia de rede é o aplicativo ConfigMaker [CON 2001] da Cisco, que permite aos administradores escolherem produtos Cisco para construir a topologia da rede.

Para se verificar se uma topologia definida é capaz de fornecer os parâmetros de QoS desejados, técnicas de simulação, como verificado antes, podem ser utilizadas. Se o comportamento simulado é diferente do desejado, a topologia da rede e os dispositivos

provavelmente devem ser alterados. Um exemplo importante de simulação de rede é a ferramenta *NS Network Simulator* [NS 2001]. O NS pode simular alguns protocolos de QoS como serviços diferenciados e MPLS. Outras soluções podem, entretanto, ser também programadas.

Para se atualizar a configuração e o *software* dos dispositivos de rede, os administradores devem utilizar aplicações automatizadas. A atualização de configuração pode ser realizada através de mensagens SNMP, interações via HTTP ou sessões Telnet. Cada dispositivo deve implementar MIBs que sejam capazes de influenciar no comportamento do dispositivo, alterando seu estado interno. Um servidor Web ou Telnet também podem ser usados. Atualmente, a configuração pode ser feita acessando-se alguns objetos da MIB-II, mas MIBs proprietárias são também utilizadas para se obter uma automação da configuração remota total [GRA 2000b]. Infelizmente, muitas MIBs relacionadas a aspectos de QoS estão ainda em fase de definição. Assim, a atualização de configuração é, em quase todos os casos, feita através de Telnet ou HTTP.

3.2 Descoberta de QoS

Diversos equipamentos de rede são atualmente carregados com inúmeras características e funcionalidades. Entretanto, não é raro encontrar, por exemplo, roteadores complexos sendo utilizados apenas para o encaminhamento de pacotes. Muitas destas características podem ser utilizadas para auxiliar no fornecimento de QoS, mas estas não são ativadas quando os administradores não conseguem lidar com tantas funcionalidades complexas.

A "descoberta de QoS" é a tarefa responsável pela procura, na rede de computadores, por características e funcionalidades que sejam capazes de auxiliar ou melhorar os serviços com QoS fornecidos. A descoberta de QoS pode ser realizada de duas principais formas:

- **Descoberta não intrusiva**. Na descoberta não intrusiva, a rede gerenciada é monitorada em pontos específicos através de processos monitores. Tais processos não introduzem pacotes, portanto não geram tráfego de gerenciamento. Assim que indícios de novos serviços com QoS forem observados, os monitores notificam o sistema de gerenciamento principal, indicando a existência na rede de uma nova funcionalidade de QoS descoberta pela observação.
- **Descoberta intrusiva**. Na descoberta intrusiva, agentes de descoberta verificam dispositivos específicos trocando informações de gerenciamento. A descoberta é mais efetiva, pois os dispositivos são explicitamente consultados e suas funcionalidades verificadas. Neste caso, entretanto, as consultas geram tráfego de gerência, o que consome banda disponível na rede.

A descoberta de QoS naturalmente é mais efetiva se mecanismos de descoberta estiverem disponíveis. Tais mecanismos podem ser simples (verificação periódica de equipamentos tentando achar MIBs relacionadas às arquiteturas de QoS) ou complexos (monitoração distribuída de rede para verificar traços de protocolo críticos, como IGMP [FEN 97] e RSVP [BRA 97]). A descoberta pode ser realizada através de analisadores

de protocolos (descoberta não intrusiva) ou mensagens SNMP enviadas aos dispositivos de interesse a procura de MIBs específicas (descoberta intrusiva). Além dos métodos "tradicionais", técnicas mais recentes podem auxiliar na descoberta. Tecnologias de agentes móveis [BIE 98], por exemplo, permitem que diversos agentes sejam enviados a uma rede à procura de mecanismos de QoS. Características como mobilidade, clonagem e colaboração entre os agentes tornam a descoberta uma tarefa verdadeiramente distribuída e libera o sistema de gerenciamento principal para a realização de outras atividades. Entretanto, mobilidade, clonagem e outras características são custosas e tornam as implementações mais complexas.

A descoberta de QoS está intimamente ligada a descoberta da topologia da rede. Duas abordagens podem ser então utilizadas para a descoberta de QoS em relação a sua interligação com a descoberta da topologia da rede:

- **Descobertas seqüenciais**. Inicialmente a descoberta da topologia é realizada de forma isolada para se determinar quais os dispositivos de rede que encontram-se ativos e qual a interconexão entre os mesmos (topologia). Em seguida, um processo de descoberta de QoS analisa os dispositivos ativos para determinar suas funcionalidades e facilidades em relação ao fornecimento de QoS.
- **Descobertas paralelas**. A descoberta da topologia e a descoberta de QoS são realizadas ao mesmo tempo. Um dispositivo recém identificado pela descoberta de topologia é imediatamente analisado pela descoberta de QoS. Mediante esta análise, o processo de descoberta de QoS indica ao processo de descoberta de topologia quais são os próximos dispositivos de interesse. Por exemplo, se o interesse é a descoberta de caminhos RSVP, o processo de descoberta de QoS só indicaria rotas com RSVP habilitado ao processo de descoberta de topologia.

Várias tecnologias podem ser empregadas neste sentido e possivelmente uma solução de descoberta de QoS eficiente inevitavelmente irá utilizar as tecnologias relacionadas a sistemas distribuídos, já que vários monitores e/ou agentes de descoberta acabam sendo empregados em redes com um número de dispositivos elevado. Soluções centralizadas, por outro lado, acabam sendo a regra atual, onde todo o processo de descoberta envolve o envio e recebimento de mensagens SNMP entre a estação de gerência e o dispositivo que está se procurando analisar. Um relaxamento dessa situação é a utilização de RMON [WAL 2000] e RMON2 [STA 96] [WAL 97], mas tais soluções acabam sendo insatisfatórias para o mecanismo de descoberta porque só informam dados sobre os equipamentos que efetivamente introduzem algum tráfego na rede, e em sua maioria estes dispositivos são os sistemas finais e não os equipamentos internos da rede.

Em relação às soluções de mercado, provavelmente a descoberta de QoS é a tarefa que mais carece de soluções. Mecanismos de descoberta de topologia já estão presentes nas plataformas de gerenciamento há algum tempo. Para a verificação dos dispositivos ativos, tais mecanismos utilizam normalmente requisições ICMP, mas diversas outras abordagens e técnicas podem ser utilizadas [SIA 98]. Entretanto, mecanismos para descoberta específica são menos freqüentes, e a descoberta de QoS é ainda menos comum.

Além disso, o tema descoberta de QoS ainda não é amplamente difundido e

investigado. Muitas vezes, sequer é interesse de fornecedores a investigação do tema. Por exemplo, o sistema HDR (*High Data Rate*) [QUA 2001] para ambientes de transmissão sem fio, em desenvolvimento pela Qualcomm [QUA 2000], fornece canais de comunicação entre 38,8Kbps e 2,4Mbps. A vazão do sistema é dependente da carga do canal e distância do ponto de acesso à rede. Devido a essa característica intrínseca, é relevante discutir a possibilidade de descobrir e fornecer informações a respeito do QoS do canal de transmissão, de maneira que a aplicação possa se adaptar ao estado do canal. No caso do sistema HDR, nenhuma informação a respeito do QoS do canal é fornecida. A Qualcomm informa que isso afetaria o modelo de camadas da arquitetura TCP/IP [COM 91] que não fornece mecanismos para obter o tipo de informação solicitada. De outro modo, as pessoas que trabalham com protocolos GPRS (*General Packet Radio Service*) [ETS 98] também indicam frustração para dispor QoS sem a presença de uma API que forneça informações de estado de canal [MIT 2000].

As pesquisas, atualmente em desenvolvimento, estão principalmente relacionadas à descoberta de QoS a partir das aplicações dos usuários [XU 2001]. O objetivo principal normalmente é a verificação do QoS fornecido pelo ambiente de comunicação e a partir disso se tentar realizar adaptações nas aplicações de modo que o QoS existente possa ser utilizado de forma mais otimizada. Assim, as aplicações verificam o estado da rede antes de iniciar a troca de informações.

Por outro lado, a descoberta de QoS por parte do sistema de gerenciamento é uma tarefa ainda mais crítica, já que é o sistema de gerenciamento o responsável por administrar os serviços com QoS existentes. Neste contexto, as pesquisas existentes são raras e normalmente associadas à descoberta de características dos dispositivos, e não ao QoS especificamente [PIN 2000].

O principal benefício da descoberta de QoS é auxiliar os administradores na definição da arquitetura de QoS a ser implantada. Com os novos dispositivos descobertos e suas correspondentes características, os gerentes possuem informações suficientes para iniciar a definição da arquitetura de QoS a ser utilizada na rede. Depois disso, a descoberta de QoS auxilia no aumento dos dispositivos com QoS catalogados. Novos roteadores com atrasos menores poderiam formar rotas com melhor QoS. Tais rotas poderiam ser então automaticamente determinadas se a descoberta de QoS encontrasse os novos roteadores instalados na rede e suas características e funcionalidades.

3.3 Manutenção de QoS

Na implantação de QoS, como visto antes, é tomada a decisão de qual arquitetura de fornecimento de QoS deve ser utilizada em uma rede particular. A correta operação da arquitetura escolhida é resultado de configurações executadas no momento de sua instalação, mas também é resultado de configurações executadas depois que a arquitetura foi implantada. Estas segundas configurações são necessárias para adaptar o comportamento da rede às constantes mudanças nos objetivos da rede gerenciada. Com a dinamicidade atual nas mudanças de tais objetivos, as configurações a serem executadas na arquitetura de QoS devem ser realizadas diversas vezes durante um curto período de tempo, como semanas e dias ou até mesmo horas e minutos.

Depois que a arquitetura de QoS é definida e implantada na rede, espera-se que o conjunto e freqüência de alterações para manutenção da arquitetura escolhida sejam

grande. Além disso, as operações de manutenção devem ser realizadas ao mesmo tempo em que os usuários da rede estão sendo atendidos, isto é, a manutenção ocorre durante a operação da rede, o que torna inevitavelmente a tarefa mais complexa. Neste trabalho, classifica-se como "manutenção de QoS" os procedimentos tomados para definir o comportamento dos serviços com QoS ao mesmo tempo em que estes serviços estão sendo oferecidos.

A manutenção de QoS visa adaptar os serviços da arquitetura de QoS aos objetivos da rede e às expectativas de seus usuários. Como dito antes, os objetivos podem mudar muito rapidamente em questão de horas ou minutos. Por exemplo, a determinação da alta prioridade de aplicações de replicações de bases de dados, entre três e cindo horas da manhã, é uma mudança de objetivos que deve ser constantemente realizada em relação aos objetivos "padrão" de uma rede.

Procedimentos freqüentemente realizados na manutenção de QoS são aqueles relacionados à classificação de tráfego, marcação e priorização de pacotes. Reserva estática de banda, gerenciamento de SLAs [MCB 96] e PBNM são também exemplos de operações de manutenção de QoS.

De todas as tarefa de gerenciamento de QoS definidas nesta tese, a manutenção de QoS é aquele executada com maior freqüência. Nela estão envolvidos três principais elementos: a rede com QoS, o administrador e seu sistema de gerenciamento, e os usuários e suas expectativas em relação ao funcionamento dos serviços com QoS.

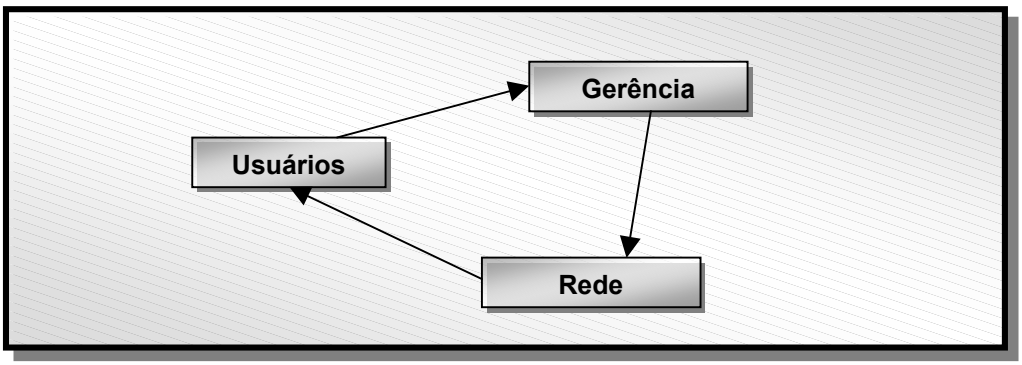

FIGURA 3.1 – Manutenção de QoS

A FIGURA 3.1 apresenta a interação entre rede, usuário e a gerência da rede. O comportamento dos serviços com QoS é observado pelos usuários que, ao perceberem um comportamento inadequado, entram em contato com a gerência da rede. O administrador deve, então, programar os dispositivos de forma que a rede comporte-se de acordo com as necessidades dos usuários. É importante notar que mesmo uma rede corretamente configurada pode não ser adequada para alguns usuários. Por exemplo, em uma rede onde a navegação Web é o serviço mais importante, provavelmente os usuários que desejam realizar videoconferência (neste exemplo, um serviço não prioritário) irão se sentir frustrados com a banda disponível. Logo, é função do administrador analisar as necessidades existentes e encontrar uma configuração de rede geral que satisfaça o maior número de usuários.

A manutenção de QoS, assim como a implantação de QoS, pode ser executada localmente ou remotamente. A intervenção local mais uma vez deve ser evita, e deve se dar preferências aos mecanismos que implementam uma manutenção remota. Tais

mecanismos podem ser constituídos de soluções padrão de configuração de equipamentos, como SNMP e Telnet. Soluções proprietárias, ainda que não preferidas, são também muito utilizadas, como por exemplo, o CLI (*Command Line Interface*) da Cisco [CIS 2001]. O IETF vem se preocupando com a questão de configuração de equipamentos de rede e produziu alguns trabalhos importantes neste sentido, ainda que muitas vezes concorrentes. Por exemplo, o grupo snmpconf (*Configuration Management with SNMP*) [SAP 2000] pesquisa a utilização do SNMP como meio de configuração [MAC 2001]. Já o grupo policy (*Policy Framework*) [HAL 2000], acredita que configurações de redes mais adequadas são aquelas realizadas através do novo protocolo COPS [DUR 2000]. O grupo rap (*Resource Allocation Protocol*) [HAH 2000], por suja vez, apóia a utilização de COPS e RSVP [HER 2000].

As soluções de mercado para a manutenção de QoS estão principalmente relacionadas com o PBNM e com o gerenciamento de SLAs. As soluções mais representativas, neste caso, são aquelas verificadas no capítulo 2. Como visto, o grande problema existente neste ponto é a falta de interoperação entre as soluções de mercado existentes, pois estas não seguem totalmente os trabalhos de tentativa de padronização desenvolvido pelo IETF.

Apesar da manutenção de QoS não cobrir apenas o PBNM, a utilização de políticas é possivelmente o mecanismos mais eficiente de atualização da configuração de equipamentos em uma rede heterogênea com muitas informações para serem tratadas. Além das políticas tratarem adequadamente o excesso de informações disponível, o uso de consumidores de políticas (elementos de tradução) permite que os métodos de acesso aos dispositivos sejam confinados internamente a estes elementos, sem que o administrador da rede tenha que se preocupar com esta questão. Como conseqüência destas vantagens alcançadas, o modelo de gerência integrada de QoS a ser apresentado na próximo capítulo desta tese utiliza políticas como o mecanismo principal para a execução da manutenção de QoS.

3.4 Monitoração de QoS

Quando uma arquitetura de QoS é implantada na rede e adequadamente mantida, respectivamente através da implantação e manutenção de QoS, espera-se que os serviços oferecidos aos usuários comportem-se de uma forma consistente. Para que a manutenção de QoS seja realizada, o administrador da rede acaba conhecendo as necessidades dos usuários e as priorizações a serem aplicadas. Como resultado, o administrador possui um conjunto de descrições de QoS passadas aos processos de manutenção que visam levar a rede a um estado adequado de acordo com seus objetivos.

Infelizmente, o QoS especificado através da manutenção de QoS pode não ser respeitado pelas estruturas da arquitetura de fornecimento de QoS, o que leva a uma situação onde o QoS real observado pode ser muito diferente do QoS especificado desejado. A diferença entre o QoS observado e o QoS especificado não pode ser maior que um limiar crítico definido. Se o QoS encontrado na rede for muito diferente do QoS esperado, as aplicações dos usuários irão apresentar degradação de desempenho, indicando que os serviços contratados não estão operando adequadamente.

Para que possa existir uma comparação entre o que foi definido e o que realmente acontece na rede em relação ao QoS, os administradores devem coletar dados

na rede e comparar com as especificações realizadas na manutenção de QoS. Assim, a monitoração de QoS é responsável por analisar o comportamento dos serviços e comparar se o QoS existente é compatível com o QoS especificado. Seguindo os trabalho de Jian et al. [JIA 2000], a monitoração de QoS deve ser capaz de determinar duas informações relacionadas:

- **QoS fim-a-fim**. Os parâmetros de QoS de segmentos específicos são importantes, mas o QoS fim-a-fim é crucial. Se o QoS fim-a-fim não estiver correto, os administradores devem ser advertidos para que o problema possa ser corrigido. A degradação do QoS fim-a-fim é a soma das degradações de cada segmento que conectam as duas extremidades de um fluxo. Se um único segmento estiver, por exemplo, introduzindo degradação, tal degradação será notada no QoS fim-a-fim.
- **Pontos de degradação**. Se for notada degradação do QoS fim-a-fim, a monitoração de QoS deve ser capaz de determinar quais os pontos na rede que influenciam em tal degradação.

Atualmente, as soluções de monitoração de QoS são capazes de satisfazer apenas o primeiro tópico. É simples verificar se existe degradação fim-a-fim utilizando-se, por exemplo, os protocolos RTP/RTCP [SCH 96]. Entretanto, identificar os pontos de degradação na rede é um procedimento mais complexo, e requer um processamento também mais complexo das informações da rede [JIA 2000].

Pode-se classificar a monitoração de QoS também em relação aos ambiente onde ela atua. As soluções de monitoração de QoS são utilizadas em dois ambientes distintos: fim-a-fim e dentro da rede.

A monitoração de QoS fim-a-fim é realizada através de elementos monitores nos sistemas finais de uma comunicação. Tais elementos são normalmente colocados dentro das aplicações que necessitam de monitoração fim-a-fim. Neste caso, não existem elementos de monitoração dentro da pilha de protocolos de rede: cada aplicação precisa implementar seus próprios procedimentos de monitoração. O protocolo de monitoração fim-a-fim mais difundido é o RTP/RTCP [SCH 96], que é utilizado, por exemplo, na ferramenta de videoconferência Vic [VIC 98].

Outra característica da monitoração fim-a-fim é o fato de que raramente existe algum tipo de notificação automática sobre as degradações ao sistema de gerência da rede. As aplicações que executam a monitoração realizam adaptações em situações de congestionamento, mas não informam a gerência da rede sobre a degradação observada. Assim, muitas vezes a gerência da rede sequer toma conhecimento de degradações, a não ser que o usuário humano entre em contato.

Já na monitoração interna à rede de computadores, o sistema de gerência toma conhecimento das degradações para que se possa identificar a localização dos pontos problemático em relação ao QoS. Diferentemente da descoberta de QoS, a monitoração de QoS é tipicamente não intrusiva, pois precisa apenas observar o comportamento dos serviços, sem introduzir nenhum tráfego na rede. A única exceção ocorre quando as degradações são observadas e o administrador da rede deve ser informado. Neste momento alguns recursos de rede são consumidor pela monitoração no tráfego das mensagens de notificação.

Atualmente, as ferramentas de monitoração de rede mais utilizadas são as arquiteturas RMON e RMON2 [STA 96]. MIBs RMON são capazes de coletar dados

importantes e alguns destes podem ser utilizados para se derivar certos parâmetros de QoS. Da mesma forma, as já citadas soluções NeTraMet [BRO 2001] e ntop [DER 2001] são exemplos de sistemas que podem ser utilizados para monitoração de QoS. Outros exemplos são NetEnforcer [ALL 2001] e VQmon [TEL 2001].

De forma geral, o mercado de monitoração apresenta um conjunto bem expressivo de ferramentas para monitoração. Alguns são capazes de gerar relatórios detalhados sobre a ocupação da rede, aplicações mais críticas e que consomem mais recursos, e as comunicações entre os sistemas finais, por exemplo. Apesar destas facilidades, observa-se uma carência em relação a soluções capazes de verificar especificamente parâmetros de QoS (tipicamente, vazão, atraso, perda e variação de atraso). Apesar da vazão ser um item bem explorado e "fácil" de ser verificado, por exemplo, através de ferramentas como o MRTG [OET 2001], os outros parâmetros de QoS são pouco abordados e requerem instrumentações nas soluções e aplicações complexas de serem realizadas, e nem sempre disponíveis aos administradores [TSY 2001].

Outro aspecto importante das soluções existentes é que a observação da rede apenas gera relatórios sobre os fluxos. Não existe uma solução que compare os fluxos observados com limiares críticos e notifiquem o sistema de gerência sobre a ocorrência de degradações. Como conseqüência, não existe uma solução de monitoração de QoS integrada à manutenção de QoS.

Por outro lado, também é importante notar que se espera que uma solução de monitoração seja capaz apenas de identificar as degradações e pontos críticos correspondentes. A correção de um degradação observada está fora do escopo da monitoração de QoS, e deve ser executada por uma entidade externa aos processos de monitoração.

3.5 Análise de QoS

Em um gerenciamento pró-ativo, os administradores de rede devem conseguir identificar problemas antes mesmo que estes ocorram. Um gerenciamento de QoS pró-ativo permitiria determinar, por exemplo, por quanto tempo um enlace WAN poderia suportar as requisições e crescentes necessidades de seus usuários. Para se alcançar um gerenciamento de QoS pró-ativo, tarefas de análise de QoS devem ser executadas.

A análise de QoS assemelha-se à monitoração de QoS, mas opera em uma escala temporal maior. Enquanto a monitoração visa verificar como se comportam os mecanismos de fornecimento de uma forma mais instantânea, a análise de QoS procura verificar o comportamento histórico dos serviços, operando na escala de dias, semanas, meses e anos.

Processos de análise de QoS poderiam utilizar, a princípio, fontes de dados diversas. Por exemplo, o comportamento histórico de um roteador poderia mostrar o número de sessões RSVP negadas devido à falta de recursos. Para tal, a análise de QoS consultaria uma MIB RSVP diretamente no dispositivo de interesse, para traçar o comportamento do serviço. Se o número de sessões negadas crescesse muito, isso indicaria que o gerente deveria atualizar os recursos da rede. Outro exemplo, seria a análise de uma operação cliente/servidor freqüente que depende de serviços com QoS. A análise da operação poderia mostrar a freqüência de degradação de QoS e os pontos

mais freqüentes dessa degradação. Neste caso, o gerente deveria analisar os pontos críticos à procura de problemas nos enlaces, ou então evitar que a operação cliente/servidor utilizasse os pontos problemáticos nas suas transações.

A fonte mais rica de informações para análise, entretanto, são os dados provenientes da tarefa de monitoração de QoS. Isso se deve ao fato de que os dados gerados pela monitoração retratam o QoS oferecido pela rede de computadores. Outras fontes de informação (por exemplo, MIBs SNMP), dificilmente seriam capazes de forneceres tais dados aos processos de análise. As informações observadas pela monitoração poderiam ser armazenadas pela análise de QoS para um estudo histórico mais completo. Neste sentido, monitoração e análise de QoS são tarefas de gerência que se complementam.

A implementação de processos de análise de rede atualmente é simples dada a quantidade e qualidade de ferramentas disponíveis. Apenas como questão de nomenclatura, as soluções hoje existentes dizem fazer monitoração da rede. Neste trabalho, entretanto, a monitoração é restrita a uma escala temporal menor em relação a análise. Assim, no contexto desta tese, as soluções disponíveis são utilizadas agora para a realização de análise e não de monitoração.

Ferramentas tradicionalmente utilizadas no gerenciamento padrão podem ser aplicadas à análise de QoS. O exemplo mais difundido de análise (ou monitoração em escalas de tempo maiores como horas, dias e meses) é o *software* MRTG [OET 2001]. O MRTG é utilizado, por exemplo, para a análise do comportamento dos enlaces do *backbone* da Internet2 [INT 2001]. Outros *softwares* desse tipo, como o Cricket [CRI 2001], podem ser utilizados da mesma forma. A grande vantagem destas soluções é que elas permitem que diversos tipos de dados, independentemente de fonte, possam ser analisados historicamente. O Cricket, por exemplo, pode ser utilizado para guardar a carga da CPU de um roteador desde que se indique como conseguir tal informação. A utilização destas ferramentas para análise de QoS é quase que direta. Basta o administrador da rede fornecer os métodos para aquisição das informações relacionadas ao QoS, e as ferramentas passam a executar a análise imediatamente. Outra vantagem, é que estas soluções são de código aberto, e possuem normalmente uma comunidade de usuário desenvolvedores extremamente expressiva.

Assim como acontece na monitoração de QoS, onde no momento da observação de uma degradação ocorre uma notificação ao sistema de gerência, é uma atribuição da analise de QoS notificar o sistema de gerência sempre que um comportamento histórico analisado se mostrar inadequado. Isso livra o administrador de rede de ficar constantemente verificando gráficos e resultados parciais de análises em andamento, deixando para os processos de análise a verificação de tais comportamentos inadequados. Infelizmente, esta funcionalidade não é amplamente encontrada nas soluções citadas anteriormente, e para sua implementação é necessário um grande número de instrumentações.

Novamente, assim como não era de responsabilidade da monitoração de QoS a resolução das degradações observadas, também não é de responsabilidade da análise de QoS a resolução dos comportamentos inadequados detectados. Assim que notificado, o administrador da rede é que deve proceder com ações na tentativa de sanar os problemas informados pela análise. Automatizações podem ser utilizadas neste caso, mas serão soluções externas à análise de QoS. Em resumo, a análise de QoS complementa a monitoração de QoS verificando, em uma escala temporal de horas, dias, meses e/ou

anos, o comportamento dos serviços de QoS, e mediante comportamentos inadequados detectados informa o sistema de gerência principal através de notificações.

3.6 Visualização de QoS

O gerenciamento baseado em dispositivos, como visto anteriormente, permite ao administrador de rede investigar cada dispositivo de interesse em particular, mas é inapropriado à gerência de QoS. As plataformas de gerência de redes comerciais, por seguirem a filosofia do gerenciamento orientado a dispositivos, apresentam as informações de gerenciamento de uma forma insatisfatória do ponto de vista da gerência de QoS.

Em uma rede com QoS, mas gerenciada através de plataformas padrão, o administrado da rede é obrigado a verificar cada dispositivo para determinar quais possuem facilidades e serviços para fornecimento de QoS. Nesta situação, a investigação acaba se tornando uma tarefa que consome muito tempo e que deveria ser substituída por um procedimento de procura automatizado.

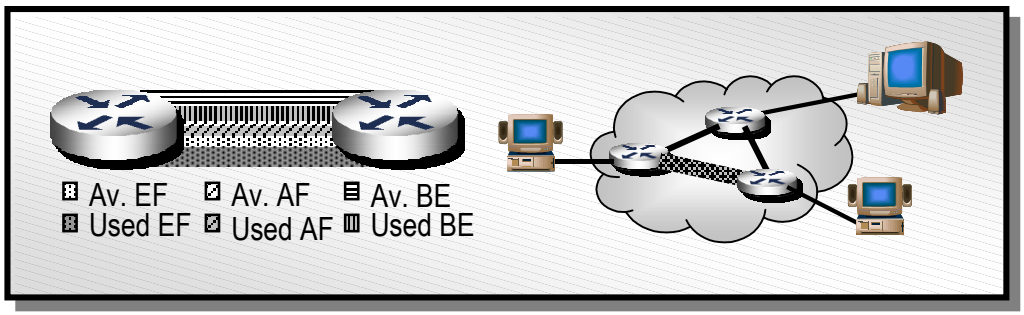

FIGURA 3.2 – Exemplos de possíveis visualizações de QoS

Assim, a visualização de QoS é uma tarefa auxiliar que ajuda os gerentes de rede a procederem com as outras tarefas relacionadas ao QoS através da disponibilização de visualizações mais adequadas dos serviços com QoS. As ferramentas de gerenciamento deveriam fornecer visualizações baseadas em características de QoS. Abaixo, são listados alguns exemplos de visualizações de QoS que poderiam facilitar a gerência de uma rede.

- **Utilização de enlace colorida**. Cada enlace mostraria, no lugar de uma linha preta comum, um conjunto de linha coloridas descrevendo a utilização do enlace por cada fluxo ou agregado (FIGURA 3.2, à esquerda);
- **Sessões fim-a-fim selecionadas**. Um mapa topológico poderia mostrar sessões fim-a-fim, selecionadas através de linha coloridas destacando as sessões. Os gerentes poderiam solicitar a visualização de sessões que correspondessem apenas a características pré-definidas;
- **Dispositivos com serviços de QoS**. Um mapa poderia mostrar os dispositivos que apresentam serviços com QoS de uma forma destacada. Cores diferentes indicariam dispositivos que implementam soluções diferentes. Por exemplo, caixas verdes poderiam indicar roteadores com suporte DiffServ; caixas vermelhas indicariam roteadores com suporte IntServ;

- **Segmentos que apresentam degradação de QoS**. Segmentos que apresentam degradação de QoS poderiam ser mostrados destacados, indicando a degradação (FIGURA 3.2, à direita).

A visualização de QoS é o resultado final de um conjunto de processamentos de informações. Antes da visualização de QoS, a informação a ser verificada deve ser recuperada e tratada. Informações localizadas em diferentes fontes devem ser comparadas e seus relacionamentos derivados. Abstrações devem ser também executadas para se descartas as informações menos importantes. Finalmente, de acordo com o desejo do administrador da rede, as informações são apresentadas de uma forma especificamente configurada.

Assim, a visualização de QoS é o resultado da: captura de informações, processamento de informações, preferências visuais do usuário (administrador da rede) e tratamento gráfico dos dados. A captura das informações de gerência normalmente é baseada em SNMP quando a fonte da informação é um dispositivo gerenciado. Entretanto, como as visualizações não se restringem apenas aos dados recuperados nos dispositivos, pode-se esperar uma coleta de informações utilizando diversas tecnologias diferentes (por exemplo, SQL, COPS, LDAP, XML, ICMP, etc.). O processamento das informações varia de acordo com o resultado final esperado das visualizações. As preferências visuais dos administradores de redes são utilizadas para informar ao processo de tratamento gráfico de dados como as informações de QoS devem ser apresentadas. Por fim, o processamento gráfico deve então utilizar tecnologias capazes de gerar gráficos consistentes com o que o usuário (administrador) solicitou e com as informações que foram efetivamente coletadas na rede.

Em relação à visualização de QoS, poucos trabalhos são desenvolvidos. O principal motivo pode ser o fato de que a comunidade de pesquisa em redes de computadores está atualmente mais preocupada com o funcionamento dos mecanismos de fornecimento de QoS do que propriamente com a visualização gráfica correspondente. Por outro lado, é argumento do autor desta tese que a visualização de QoS, apesar de ser a última tarefa de gerência de QoS definida, seja talvez a mais importante delas, pois é a que efetivamente informa ao administrador da rede o comportamento da rede gerenciada. Sem a visualização não seria possível aplicar, por exemplo, configurações na rede e saber o resultado efetivo das operações.

As tecnologias de apresentação das informações graficamente são variadas. Com a atual tendência de utilização da gerência baseada na Web, as possibilidades de visualização herdaram todas as técnicas gráficas de visualização de informações embutidas em conteúdo Web. O gerenciamento baseado na Web vem sendo investigado já há alguns anos, mas um primeiro movimento significativo ocorreu apenas em 1996, quando surgiram as primeiras especificações WBEM (*Web-Based Enterprise Management*) [DIS 2001] como parte da solução DMTF (*Distributed Management Task Force*). Martin-Flatin et al. produziu também investigações importantes em gerenciamento baseado na Web a ponto de propor uma plataforma totalmente baseada em Java como solução [MAR 99]. XML (*Extensible Markup Language*) usado na gerência de redes também ganhou importância quando o XML foi incorporado à solução WBEM. A partir de uma abordagem diferente, John et al. também propôs, em 1999, o sistema XNAMI [JOH 99] que utilizava XML para reduzir o tráfego de gerência permitindo a transferência de MIBs entre dispositivos gerenciados e estações de gerência. Infelizmente, nenhuma das soluções baseadas na Web anteriores levam em

conta as questões de QoS diretamente, o que neste caso não resolve a problemática da visualização de QoS.

3.7 Evolução da gerência de QoS e interação entre as tarefas de gerência

Nas seções anteriores foram apresentadas as tarefas de gerência de QoS. As tarefas naturalmente não são executadas separadamente e interações entre tais tarefas são esperadas. Além disso, o momento em que as tarefas devem estar "ativas" é importante para verificar quando cada uma deve ser iniciada, finaliza ou mantida. Nesta seção é inicialmente apresentada a execução das tarefas em relação à evolução da gerência da rede. Após, são discutidas as interações entre as tarefas durante esta evolução da gerência e como isto influencia a rede gerenciada.

A FIGURA 3.3 apresenta a execução das tarefas de gerência de QoS em relação à evolução da gerência da rede. Supõe-se inicialmente que exista uma rede não definida (sem equipamentos) ou uma rede definida mas sem uma arquitetura de gerência implantada (por exemplo, uma rede IP padrão). Neste momento o administrador procede então com a implantação de QoS. Com a utilização de ferramentas capazes de descrever graficamente a rede a ser gerenciada, e com o uso de simulações, o administrador decide qual arquitetura de fornecimento de QoS é mais adequada ao seu ambiente. A seguir, a arquitetura escolhida deve ser disponibilizada, como visto anteriormente, através da inclusão de novos equipamentos, atualização de *software*, etc. A implantação de QoS é então encerrada quando a arquitetura de fornecimento de QoS escolhida estiver totalmente disponibiliza aos usuários da rede.

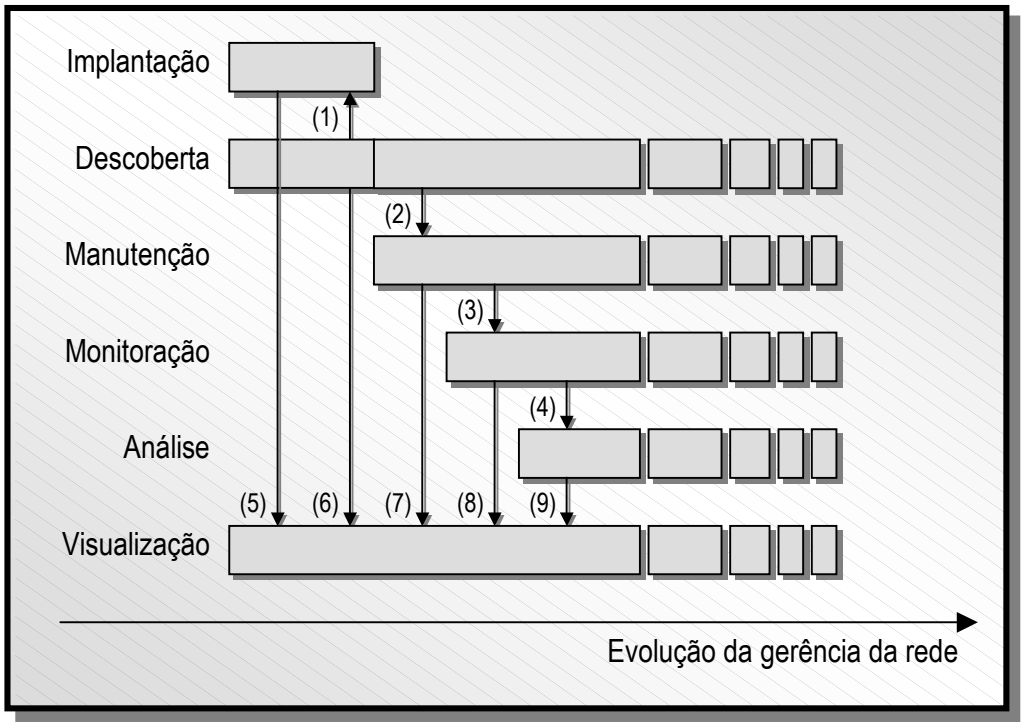

FIGURA 3.3 – Evolução da gerência da rede em relação aos aspectos de QoS

A descoberta de QoS é iniciada em momentos diferentes de acordo com a rede gerenciada. Se a rede ainda não existir, a descoberta de QoS inicia após a implantação

de QoS. Se a rede já existir, então a descoberta de QoS inicia junto com a implantação de QoS. Os novos dispositivos descobertos, no segundo caso, são utilizados pelo administrador da rede para determinar a topologia da rede e quais as facilidades de QoS já existentes na rede atual. A descoberta de QoS permanece uma tarefa ativa indefinidamente para que novos dispositivos colocadas na rede sejam descobertos automaticamente, evitando que o gerente da rede tenha que cadastrá-los no sistema de gerência manualmente. Isso permite, por exemplo, que uma rota com melhor QoS criada com a colocação de um novo roteador na rede seja automaticamente identificada pela descoberta de QoS.

Assim que a arquitetura de QoS for implantada, inicia-se a manutenção de QoS. Esta tarefa também permanece ativa indefinidamente para que as mudanças nos objetivos da rede sejam executadas sempre que necessário através da reconfiguração de seus dispositivos.

Partindo do fato de que a arquitetura de QoS pode não atender adequadamente às especificações de QoS realizadas pela manutenção, o gerente da rede passa a proceder, pouco tempo após o início da manutenção, com o monitoração de QoS. As definições realizadas na manutenção de QoS são repassadas à monitoração de QoS que verifica, junto à rede, se o QoS acordado está sendo respeitado. Novamente, a monitoração de QoS permanece ativa indefinidamente enquanto existirem definições de QoS realizadas na manutenção. Um fato interessante, entretanto, é que a monitoração de QoS pode ser utilizada também em uma rede sem QoS! Neste caso, uma definição hipotética de QoS deve ser fornecida para que os processos de monitoração possam comparar os fluxos observados com o QoS hipotético. Sem o fornecimento desta definição, o processo de monitoração se tornaria um processo de análise simples, e seria classificado como análise de QoS e não monitoração de QoS.

A análise de QoS passa a ser executada após o início da monitoração de QoS. Isso é necessário para que as informações provenientes do processo de monitoração estejam disponíveis para serem analisadas. Na prática, a análise de QoS poderia ser ativada antes, por exemplo junto com a manutenção de QoS (já que é possível recolher dados de análise diretamente nos dispositivos). Entretanto, as análises realizadas serão mais pobres se comparadas com as análises feitas já com as informações da monitoração de QoS disponíveis. O processo de análise também permanece ativo indefinidamente, acompanhando as tarefas de manutenção e monitoração de QoS.

Por fim, a visualização de QoS é a tarefa mais abrangente de todas porque inicia, independentemente da rede existir ou não, junto com a implantação de QoS e permanece ativa indefinidamente. Isso se deve ao fato de que todas as tarefas de gerência de QoS precisam de uma representação gráfica para serem utilizadas pelo administrador da rede. Logo, enquanto existir qualquer outra tarefas de gerência em execução, paralelamente existirá também a visualização de QoS. Isso é representado na FIGURA 3.3 pelas setas verticais que partem de todas as tarefas de gerência e chegam à visualização de QoS (setas de 5 a 9).

As outras setas da FIGURA 3.3 representam outras interações entre as tarefas. A descoberta de QoS auxilia o gerenciamento em dois importantes momentos: na implantação de QoS (seta 1) e na manutenção de QoS (seta 2). Na implantação de QoS, as novas características descobertas de um dispositivo podem ser utilizadas para se decidir sobre a arquitetura de QoS a ser implantada. Na manutenção de QoS, a descoberta pode reportar, enquanto a arquitetura de QoS está em operação,

equipamentos recentemente adicionados à rede que podem fornecer um melhor QoS em relação ao QoS já existente.

As definições de comportamento de rede criadas na manutenção de QoS devem ser utilizadas pela monitoração de QoS para comparações (seta 3). Assim, os dados de entrada dos processos de monitoração de QoS são as definições criadas e implantadas pela manutenção. Tipicamente, o uso de políticas neste contexto serve para descrever o QoS desejado, e implantá-lo na rede. Serve também para que a tarefa de monitoração possa verificar se as políticas definidas estão realmente se comportando de forma adequada.

Por fim, a análise de QoS pode ser realizada tomando como partida dados recuperados de diversas fontes, como visto anteriormente. Entretanto, os dados para análise mais ricos são aqueles conseguidos junto aos processos de monitoração de QoS. Assim, o resultado da monitoração é dado de entrada dos processos de análise de QoS (seta 4). Novamente, o resultado da análise de QoS é então apresentado ao administrador da rede através das facilidade de visualização (seta 9).

Além das interações diretas entre tarefas de gerência de QoS apresentadas na FIGURA 3.3, outras interações indiretas acabam acontecendo com a intervenção do administrador da rede. Por exemplo, da análise de QoS o administrador pode tomar a decisão de restringir o escopo de um política que consome muitos recursos de rede. A descoberta de um novo equipamento poderia fazer com que o administrador tomasse, por precaução, a decisão de utilizá-lo para fluxos mais críticos apenas após a confirmação, através da análise de QoS, de que o equipamento apresenta comportamento suficientemente estável.

3.8 Análise sobre integração de tarefas de gerência de QoS

Neste capítulo foi apresentada a classificação dos aspectos envolvidos na gerência de QoS em tarefas de gerenciamento de QoS. Além disso, a seção anterior apresentou também a execução destas tarefas em relação à evolução de gerência de QoS de uma rede, bem como o relacionamento existente entre as tarefas.

Algumas soluções para a execução das tarefas de gerência foram apresentadas, o que leva a conclusão de que tais tarefas já podem ser executadas hoje utilizando tais soluções. De fato, muitas das tarefas são realmente executadas utilizando-se ferramentas de mercado. Apenas como exemplo, o gerenciamento de SLAs (manutenção de QoS) é uma realidade entre domínios administrativos diferentes, e é realizado com ferramentas específicas.

Apesar das tarefas já poderem ser executadas e seguirem a evolução da gerência de QoS, como apresentado principalmente através da explicação da FIGURA 3.3 da seção anterior, a integração entre as tarefas não é uma realidade. Na prática, analisando-se as soluções existentes, as linhas verticais indicativas das interações entre as tarefas não existem nos ambientes reais de gerenciamento. Como conseqüência, o administrador da rede acaba "implementando" a integração indiretamente, utilizando o retrabalho como ferramenta de integração. O exemplo mais claro dessa situação é verificado em relação à integração da manutenção e monitoração de QoS. Atualmente, o administrador especifica inicialmente como os fluxos devem ser priorizados e programa a rede (via manutenção) para que esta execute as priorizações. Depois, os mesmos

fluxos definidos na manutenção são novamente definidos nos processos de monitoração, para se verificar sua consistência. Nesta situação, a integração permitiria que uma única definição de fluxos fosse utilizada na programação da rede (manutenção) a na observação da mesma (monitoração). Logo, o retrabalho (a dupla definição de fluxos) foi utilizado para que as ferramentas de manutenção e monitoração trabalhassem de forma integrada.

Obviamente, a utilização de retrabalho não é ideal porque o tempo gasto nas atividades é grande. Assim, existe a necessidade de integração real entre as tarefas de gerência de QoS, para que as linha verticais da FIGURA 3.3 passem a ser reais.

4 Modelo proposto para gerência integrada de QoS

Os capítulos anteriores apresentaram o problema da falta de integração das soluções de gerência de QoS (capítulo 2) e a classificação destas soluções em tarefas de gerenciamento de QoS (capítulo 3). Este capítulo apresenta a segunda contribuição da tese: a definição de um modelo para a gerência integrada de QoS.

Inicialmente são apresentados os principais requisitos que guiaram a definição do modelo. A seguir, o modelo propriamente dito é apresentado e seus elementos discutidos separadamente em seções específicas. Serão então discutidas questões sobre os protocolos e a localização dos elementos do modelo na rede de computadores. Por fim, dois exemplos de utilização do modelo em ambientes hipotéticos encerram o capítulo.

4.1 Requisitos do modelo

O requisito básico do modelo de gerência integrada de QoS apresentado a seguir, é a necessidade de **integração das tarefas de gerência de QoS** definidas do capítulo 3. Assim, o ponto principal é que o modelo deve ser capaz de fornecer facilidades que permitam a execução, de forma integrada, da implantação, descoberta, manutenção, monitoração, análise e visualização de QoS.

Além de permitir uma gerência integrada, **o modelo deve ser genérico** de forma que possa ser aplicado em redes com tecnologias diferentes. Por exemplo, o modelo deve implementar a gerência de QoS em redes ATM assim como em redes IP. A rede em que o modelo é aplicado obviamente deve possuir algum nível de suporte ao fornecimento de QoS, já que, como visto anteriormente, o adequado funcionamento dos serviços é resultado da existência da gerência de QoS e de uma arquitetura de fornecimento de QoS. A ausência de um dos elementos inibe o correto funcionamento dos serviços.

O modelo definido, além de fornecer suporte a todas as tarefas de gerência de QoS, **deve ser aberto** para permitir futuras expansões. A incorporação de novas funcionalidades deve ser realizada de forma facilitada, para que o administrador da rede, desenvolvedores de soluções de gerência, e fornecedores de dispositivos sejam incentivados a estender o modelo de acordo com suas necessidades específicas.

Além de aberto, o modelo **deve ser flexível**, de forma a operar em ambientes distintos, com requisitos de operações variados. Por exemplo, deve poder ser capaz de se adaptar aos requisitos de uma rede que necessita de tráfego multimídia da mesma forma que se adapta a uma rede onde a manutenção distribuída de bases de dados de comércio eletrônico é a tarefa mais crítica. O modelo deve também **ser escalável**, para operar em redes de proporções diferentes, progressivamente maiores. A introdução de novos usuários, serviços, equipamentos e sub-redes inteiras não deve afetar substancialmente o comportamento do modelo de gerência.

Por fim, o modelo **deve ter uma característica informativa** sobre o estado da rede, sem se preocupar tanto com a automática resolução de problemas relacionados ao

QoS. Neste contexto, admite-se que a automatização de tarefas de gerência é realizada em um nível superior ao modelo e que utiliza as informações apresentadas pela solução para tomar decisões automáticas de gerência. Este último requisito é importante para definir mais exatamente o escopo da solução, principalmente em relação às questões que envolvem o controle de QoS (discutido na introdução). Quando o tempo de resposta aos problemas torna-se crítico, o controle de QoS deve ser utilizado. Isso faz com que os elementos de uma solução de gerência passem a comunicar-se mais entre si, e menos com o administrador da rede, que acaba, em muitos casos, sequer envolvendo-se na resolução de problemas. A automatização das tarefas pelo modelo acabaria levando a uma nova solução de controle de QoS. Entretanto, a proposta aqui apresentada está mais preocupada com a solução de problemas que requerem a intervenção humana, e isso acaba exigindo um modelo com características mais informativas.

4.2 Modelo geral

Tendo em vista os requisitos da seção anterior, o modelo de gerência integrada apresentado na FIGURA 4.1 é definido.

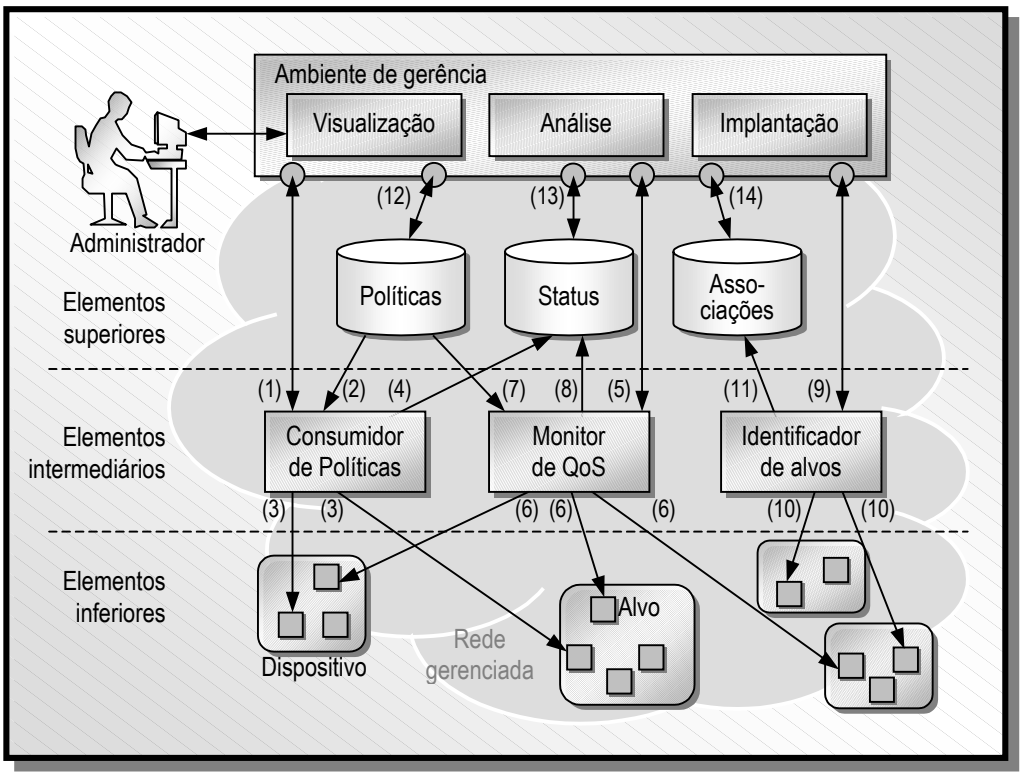

FIGURA 4.1 – Modelo de gerenciamento integrado de QoS

O modelo é baseado nos trabalhos desenvolvidos pelo IETF para a definição de estruturas e requisitos gerais de um sistema de gerenciamento baseado em políticas [MAH 2000]. O modelo do IETF, entretanto, preocupa-se apenas com as questões relacionadas às políticas. A abrangência do modelo do IETF está relacionada ao fato das políticas poderem ser utilizadas não apenas no gerenciamento de QoS, mas também no gerenciamento de segurança, por exemplo. O modelo proposto por esta tese de doutorado, por outro lado, está focado exclusivamente nas questões relacionadas ao

QoS, e como tal, relaciona-se com o modelo do IETF apenas neste aspecto. Restringindo-se às questões de gerência de QoS, nesta proposta o modelo do IETF é expandido para suportar não apenas a utilização de políticas mas também o uso de outras estruturas capazes de cobrir as seis tarefas de gerência de QoS definidas no capítulo 3.

O modelo proposto define três conjuntos de elementos em camadas sucessivas (da camada inferior para a cada superior): elementos inferiores, intermediários e superiores. Todos os elementos estão localizados na rede gerenciada e interagem entre si para a implementação da gerência integrada de QoS (a localização exata de cada elementos é discutida na seção 4.9). Os elementos inferiores são constituídos de dispositivos e alvos associados. Os elementos intermediários são três: consumidor de políticas, monitor de QoS e identificador de alvos. Por fim, os elementos superiores são em maior número, sendo constituídos pelas bases de dados e pelo ambiente de gerência. Este último, ambiente de gerência, possui internamente ainda três outros elementos: os processos para implantação, análise e visualização de QoS.

A FIGURA 4.1 apresenta os três grupos de elementos assim como as comunicações entre os mesmos e com o administrador da rede. As comunicações são representadas por setas numeradas, descritas nas próximas seções. Todos os elementos do modelo proposto serão agora detalhadamente abordados e ao final do capítulo dois exemplos de utilização do modelo são apresentados.

4.3 Dispositivos e alvos

Os alvos são os elementos ativos que tornam possível o fornecimento de QoS pela rede de computadores. Cada dispositivo de rede pode possuir diversos alvos. Por exemplo, em um roteador, cada interface do roteador é um alvo. Assim, os alvos são os elementos finais que efetivamente implementam a arquitetura de fornecimento de QoS escolhida pela tarefa de implantação de QoS.

O escopo de um alvo corresponde ao seu ambiente de atuação. Se um alvo atua especificamente em uma interface, por exemplo, como acontece com as disciplina de filas, então cada disciplina de filas de cada interface de um roteador será um alvo. Entretanto, se um alvo atua de forma mais global, afetando o comportamento geral de um roteador, então o alvo nesse caso será único para o roteador. Classificadores de pacotes que não fazem distinção de interface são exemplos de alvos que atuam globalmente. Em um roteador com um processo de classificação, 4 interfaces e 4 disciplinas de filas diferentes por interface, existirão 16 disciplinas no total, isto é, 16 alvos do tipo “disciplina de filas” e 1 alvo do tipo “classificador de pacotes”.

Cada alvo existente deve ser adequadamente classificado de acordo com as operações que executa. Assim, cada alvo possui um tipo que descreve as capacidades do alvo. Exemplos de tipos de alvos são: disciplinas de filas com diferenciação (*priority queuing* [LIN 91], *Weighted Fair Queuing* [QUA 2000a]), processos de conformação de tráfego, processos de policiamento de tráfego, etc. A classificação dos alvos em tipos é importante para que o administrador da rede consiga selecionar os alvos adequados que possam suportar as operações desejadas. Por exemplo, não faria sentido tentar aplicar uma programação de priorização de tráfego em processos de policiamento de fluxos (já que o policiamento não prioriza tráfego, mas descarta pacotes de fluxos mal

comportados). Da mesma forma, não faria sentido aplicar operações de conformação de tráfego em alvos responsáveis por classificação de agregados (já que a classificação apenas identifica a que disciplina de filas um pacote deve ser encaminhado, e não se deve haver algum tipo de conformação do mesmo).

O administrador de rede tem acesso indireto aos alvos através de interações com os elementos intermediários. As interfaces de comunicação entre os elementos intermediários e os alvos (FIGURA 4.1, setas 3, 6 e 10) são dependentes dos dispositivos onde os alvos se encontram, e diferentes protocolos devem ser utilizados para se acessar alvos diferentes. Um roteador do fornecedor A, por exemplo, poderia ser acessado via Telnet, enquanto um outro roteador de um fornecedor B poderia ser acessado via COPS.

4.4 Consumidores de políticas

Os consumidores de políticas são os responsáveis por implantar nos alvos as políticas definidas pelo administrador de rede na tarefa de manutenção de QoS. Cada consumidor, quando solicitado a proceder com a implantação de uma política, traduz as definições em alto nível da política para instruções específicas no dispositivo, de forma a programar as operações do alvo destino. Outro modo de funcionamento de um consumidor de políticas é chamado de *outsourcing*. Neste caso, o consumidor é consultado pelo alvo toda vez que uma nova decisão deve ser tomada baseada nas políticas vigentes. Por exemplo, o controle de admissão aplicado em uma solicitação RSVP pode ser descrito em políticas que são consultadas nos consumidores sempre que uma nova tentativa de estabelecimento de sessão for solicitada a um alvo.

Durante a implantação de uma política, o consumidor é também responsável por verificar o sucesso desta política junto ao alvo. Se a política não puder ser implantada devido a uma falha ou falta de recursos no alvo, o consumidor notifica o administrador da rede enviando uma mensagem ao ambiente no usuário (FIGURA 4.1, seta 1, no sentido de baixo para cima). (Uma atualização da base *status* também ocorre, como mostrado pela seta 4 e apresentado mais à frente). Cada dispositivo de rede possui uma certa capacidade máxima de processamento e armazenamento. Um exemplo de falha na implantação de uma política é a tentativa de programar uma disciplina de filas de uma interface de um roteador que utiliza prioridades através da inclusão de uma nova regra que não pode ser comportada, ou pela falta de memória ou porque o número máximo de regras permitidas já foi alcançado.

Um consumidor de políticas pode implantar políticas em diversos alvos ao mesmo tempo (setas 3). Da mesma forma, cada dispositivo pode estar associado a vários consumidores de políticas ao mesmo tempo. Apesar disto, cada alvo deste dispositivo só pode estar associado a um único consumidor. A associação de vários consumidores a um mesmo alvo complicaria o modelo, além de exigir a necessidade de contenção dos consumidores em relação ao acesso ao alvo comum para evitar inconsistências. Entretanto, na implementação de tolerância à falhas, mais cópias de um mesmo consumidor pedem ser admitidas e associadas a um mesmo alvo como redundância, mas apenas uma das cópias seria o consumidor ativo ou corrente.

A transferência de políticas entre ambiente de gerência e consumidor de políticas pode ocorrer de duas formas distintas: através do modelo *push* ou através do modelo

pull [FLA 99]. No modelo *push* o ambiente de gerência recupera uma política a ser implantada de um repositório de políticas (no caso do modelo proposto, o repositório de políticas corresponde à base de dados *políticas*) e repassa a mesma diretamente ao consumidor (seta 1, no sentido de cima para baixo). Já no modelo *pull*, o ambiente de gerência repassa apenas a localização da política (por exemplo, através de uma URL), e o consumidor de políticas é responsável por acessar o endereço informado e recuperar a política diretamente (seta 2). Neste caso, o endereço repassado corresponderia ao repositório de políticas associado (base de dados *políticas*, novamente). O IETF sugere ainda como solução para a implantação de políticas a constante monitoração, por parte dos consumidores, do repositório de políticas para a verificação de inclusão de novas definições. Entretanto, esta forma introduz um número grande de mensagens na rede, além de consumir recursos de processamento desnecessários nos consumidores e no repositório de políticas. A utilização de mecanismos de notificação entre repositório e consumidores também é uma alternativa, mas tais mecanismos são mais raros de serem encontrados, e são satisfatoriamente substituídos pelos modelos *push* e *pull* anteriores, que requerem a presença do ambiente de gerência. Como este ambiente inevitavelmente estará sempre presente na coordenação das operações, os modelos *push* e *pull* são suficientes e eficientes na transferência de políticas.

4.5 Monitores de QoS

As políticas implantadas em uma rede podem não se comportar de acordo com sua definição. O QoS resultante de uma implantação de política pode ser diferente do QoS especificado na política. Políticas críticas devem ter seu comportamento monitorado. O elemento responsável por executar tal monitoração é o monitor de QoS. O administrador de rede define quais as políticas que devem ser verificadas e então monitores de QoS associados aos alvos que implementam tais políticas são ativados (seta 5, no sentido de cima para baixo).

Os monitores de QoS acessam a definição das políticas também através dos modelos *push* e *pull.* No primeiro caso o ambiente de gerência transfere a política a ser monitorada diretamente (seta 5, no sentido de cima para baixo). No segundo caso, o monitor de QoS recupera a política de interesse na base de dados de políticas (seta 7). Os alvos associados à política são monitorados (setas 6) e o comportamento observado é comparado com o QoS especificado na definição da política. Se degradações são percebidas, o monitor de QoS notifica o administrador da rede enviando mensagens especiais ao ambiente de gerência (seta 5, no sentido de baixo para cima). (Novamente, como será discutido à frente, a base *status* também é atualizada na operação representada pela seta 8).

Assim como acontece com os consumidores de políticas, cada monitor de QoS possui capacidade de tradução de políticas e conhece o método de acesso ao alvo de interesse (por exemplo, SNMP, Telnet, etc.). Entretanto, as ações resultantes da tradução devem ser capazes de "ler" o alvo associado, e não configurar tal alvo, como acontece nos consumidores.

Um monitor de QoS também pode estar associado a diversos alvos (setas 6), monitorando seus comportamentos ao mesmo tempo. Por outro lado, quanto maior o número de alvos associados a um único monitor, maior o processamento realizado pelo monitor, o que pode tornar a identificação de degradações, e correspondente notificação,

mais lentas. Cada dispositivo pode possuir vários monitores associados (por exemplo, diversos monitores acessando objetos diferentes de uma MIB QoS). Diferentemente dos consumidores, não só um dispositivo pode ter vários monitores, mas também cada alvo do dispositivo pode estar associado a vários monitores diferentes. Isso é possível porque os monitores de QoS executam na prática apenas operações de leitura nos alvos (não gerando inconsistência), enquanto os consumidores executam também operações de escrita (se vários consumidores pudessem programar um alvo ao mesmo tempo, situações de inconsistência poderiam surgir).

4.6 Identificadores de alvos

Em redes com um número extenso de equipamentos, a procura em cada dispositivo para a identificação de seus alvos é uma tarefa que consome muito tempo. Além disso, para novos dispositivos colocados em uma rede já operante, os alvos de tal dispositivo devem ser catalogados para que possam ser usados. Por fim, se a descoberta de QoS é uma das tarefas de gerenciamento a serem suportadas, um esquema para identificação automática de alvos deve ser fornecido.

Os identificadores de alvos são os elementos do modelo proposto com a função de procurar na rede por novos alvos. Cada identificador de alvo reconhece, pelo menos, um tipo específico de alvo, usando um algoritmo específico de identificação. Por exemplo, um identificador de alvos de serviços diferenciados é aquele que procura dentro dos roteadores pela existência de priorização baseada no valor do campo DS (antigo campo ToS) [NIC 98]. Para tal, o identificador de alvos em serviços diferenciados pode abrir uma sessão Telnet ou verificar a existência de uma MIB DiffServ. Assim, um identificador de alvos deve internamente possuir o conhecimento do que deve ser procurado (por exemplo, MIB DiffServ, configurações RSVP, etc.), como acessar um alvo (por exemplo, via SNMP ou Telnet) e como procurar (que heurística de procura deve ser utilizada). Em relação a este último aspecto, assume-se que um universo de dispositivos de rede já foi identificado, mas suas características em relação ao QoS ainda não. Isso simplifica o modelo, além de ser um pressuposto factível, já que ferramentas para descoberta padrão (sem se importar com os aspectos de QoS) são amplamente utilizadas e divulgadas e poderiam facilmente ser utilizadas no modelo proposto.

Visto que cada alvo possui diferentes características, os identificados de alvos são também responsáveis por classificar novos alvos descobertos na rede atribuindo-lhes um identificador que descreva seu tipo. Por exemplo, disciplinas de filas WFQ, *custom queuing* e *priority queuing* seriam classificadas como “disciplinas com diferenciação de tráfego”. Classificadores em serviços integrados (baseados em endereço de rede origem e destino, porta de aplicação origem e destino, e protocolo de transporte) e classificadores em serviços diferenciados (que ainda utilizam o campo DS do cabeçalho IP) seriam do tipo “classificadores de pacotes”.

Em relação à operação dos processos de descoberta, o administrador define parâmetros de funcionamento que são repassados aos identificadores de alvos (seta 9, no sentido de cima para baixo). Entre os parâmetros de descoberta encontram-se os tipos de serviços de interesse e o universo de dispositivos conhecidos (já descobertos pelos processos padrões de descoberta de topologia citados anteriormente). Quando um novo alvo é identificado (setas 10), algumas associações devem ser realizadas. A mais óbvia é

associar o alvo descoberto ao seu dispositivo. Associar o alvo a um tipo de alvos, como apresentado anteriormente, também é necessário. Os alvos descobertos, juntamente com as associações realizadas, são então armazenados (seta 11) em um repositório de alvos e associações (no modelo, tal repositório é a base *associações* apresentada mais à frente).

O ambiente de gerência também deve ser notificado quando novos alvos são encontrados (seta 9, no sentido de baixo para cima). Duas abordagens podem ser utilizadas: notificações constantes ou notificações em bloco. As notificações constantes são enviadas ao ambiente de gerência sempre que um novo alvo for encontrado. Entretanto, para dispositivos que possuem diversos alvos, a notificação constante pode gerar um volume de tráfego considerável. Neste caso, a notificação em bloco é mais adequada porque só é enviada ao ambiente de gerência depois que um bloco de alvos foi encontrado. Um bloco de alvos pode ser o conjunto de alvos encontrados em um único dispositivo, ou todos os alvos encontrados em diversos dispositivos, ou ainda todos os alvos de todos os dispositivos de uma rede consultada. Um bloco de alvos poderia ser também constituído por um único alvo, o que torna a descoberta constante um caso particular da descoberta em bloco. A definição de um bloco de descoberta é dependente de implementação, mas o ideal é que o ambiente de gerência fosse flexível o suficiente para implementar vários tamanhos de blocos diferentes.

4.7 Bases de dados

O modelo de gerência proposto sugere a utilização de três bases de dados diferentes no gerenciamento: *políticas, status* e *associações*. A base de políticas implementa um repositório onde as políticas definidas pelo administrador são armazenadas (seta 12, no sentido de cima para baixo). As políticas armazenadas podem também ser acessadas para edição ou remoção do repositório (seta 12, no sentido de baixo para cima). Cada política armazenada possivelmente será pouco acessada para modificações, o que permite a utilização, na implementação da base, de soluções que permitam várias leituras (quando uma política é implantada em um dispositivo, por exemplo) e proporcionalmente poucas edições (quando uma política é alterada ou removida). O IETF, por exemplo, indica a utilização de um sistema de diretórios como o LDAP na implementação do repositório de políticas.

Quando uma política é implantada em um alvo, deve ser estabelecido um relacionamento entre a política e tal alvo. Da mesma forma, o *status* da política implantada deve ser uma informação acessível para que o administrador da rede possa verificar como a política implantada está se comportando. Assim, além das informações que definem as políticas, outras informações também devem ser fornecidas para que o sistema baseado em políticas opere adequadamente. Como conseqüência, além do repositório de políticas, devem existir outras bases de dados responsáveis por armazenar os dados complementares.

Todos os alvos, consumidores de políticas, monitores de QoS e identificadores de alvos são registrados na base de associações. A base de associações armazena também as informações sobre os relacionamentos entre estes elementos, e entre políticas e alvos. Por exemplo, um alvo chamado A pode utilizar: um consumidor de políticas B para tradução e implantação; um monitor de QoS C para verificar a performance de uma política aplicada; e um identificador de alvos D para descobrir possíveis novas características do alvo A. Na prática, a base de associações desempenha, de forma geral,

as mesmas funções que uma base de um sistema de gerência padrão desempenha. Com solicitações adequadas à base (seta 14), o ambiente de gerência pode derivar a topologia da rede gerenciada, soluções de QoS implantadas, e características dos dispositivos.

A base associações, além de ser utilizada pelo ambiente de gerência, é também acessada pelos identificadores de alvos para registrar novos alvos e dispositivos encontrados (seta 11). Comparativamente, a base de associações possui uma natureza de "escrita" bem maior que a base de políticas. Toda vez que uma política deve ser aplicada em um alvo, por exemplo, uma operação de escrita é realizada na base associações para estabelecer um novo relacionamento entre a política aplicada e o alvo. Isso, entretanto, não gera nenhuma operação de escrita na base de políticas.

De natureza ainda mais "alterável" são aqueles dados utilizados para representar o estado de uma política implantada e/ou monitorada. Para tal, a base de *status* é definida separadamente da base de políticas e de associações. Os monitores de QoS e os consumidores de políticas alteram os dados da base *status* sempre que uma política implantada tem seu estado alterado (setas 4 e 8). Assim, a base *status* liga as informações armazenadas na base *políticas* (as políticas) com as informações armazenadas na base *associações* (os alvos) e apresenta novas informações em relação a este relacionamento: o estado de uma política implantada. Tal estado pode então ser monitorado pelo ambiente de gerência sempre que necessário (seta 13). Como as informações armazenadas são de natureza mais "alterável", a utilização de uma implementação baseada em serviços de diretórios (possivelmente utilizada na implementação da base de políticas), por exemplo, não se mostra como a melhor alternativa. Neste caso, um sistema que suporte um grande número de alterações dos dados é preferível, assim como acontece com a base de associações.

Funcionalmente, as três bases de dados são definidas separadamente, mas na prática as suas implementações poderiam ser realizadas em um mesmo sistema final, ou ainda em um mesmo sistema de banco de dados. Isso permitiria uma centralização das informações de gerência, facilitando a análise de informações correlacionadas, mas em bases funcionalmente diferentes. Entretanto, a centralização pode não ser adequada se o ambiente gerenciado contiver muitos dispositivos, ou se questões de segurança forem importantes no ambiente gerenciado (dados em bases de máquinas diferentes estão protegidos da inoperância das máquinas que contém outras bases). Como dito anteriormente, o IETF indica a utilização de LDAP como solução para o armazenamento dos dados relacionados a um sistema de gerência baseado em políticas. Entretanto, o próprio IETF admite que do conjunto de informações existentes, possivelmente apenas as políticas (que são as informações menos "alteráveis" do sistema) devem realmente utilizar um serviço de diretórios. Para as outras informações é recomendado o uso de um sistema de gerência de banco de dados convencional capaz de suportar operações de escrita concorrentes.

4.8 Ambiente de gerência

O ambiente de gerência é o componente lógico do modelo que coordenada as atividades de todos os outros componentes. É o ponto central de processamento e responsável por ativar e desativar operações de gerência utilizando serviços dos demais componentes. É também o ponto através do qual o administrador da rede interage com o modelo e verifica o estado da rede gerenciada e procede com as atividades de gerência

de QoS.

Internamente ao ambiente de gerência, é encontrado o suporte para as atividades de visualização, análise e implantação de QoS. A visualização de QoS fornece a interface gráfica de usuário ao administrador da rede, permitindo a inclusão de dados, edição e remoção. Permite a verificação do estado dos equipamentos, políticas implantadas e políticas colocadas no repositório. Permite também a definição de regras para descoberta de topologia e QoS e o acesso ao resultado da monitoração e análise de QoS. Enfim, os processos de visualização permitem a interação do administrador da rede com todo o sistema de gerência.

O suporte à análise de QoS é encontrado junto ao ambiente do usuário porque esta tarefa não necessita de uma interação muito grande com os dispositivos analisados. Como a escala temporal da análise de QoS é maior que a escala temporal da monitoração de QoS, a colocação dos processos de análise "longe" dos dispositivos e serviços analisados não representa um problema. Além disso, um acesso mais eficiente aos dados resultantes da análise é possível porque o processo de visualização de QoS encontra-se "próximo" ao processo de análise.

A tarefa de implantação de QoS também encontra suporte junto ao ambiente de gerência por se tratar de uma tarefa mais de planejamento do que uma tarefa de interação direta com a rede. Além disso, muitas vezes a rede a ser gerenciada sequer existe (ou não apresenta uma arquitetura de QoS associada) o que faz o ambiente de gerência o local mais adequado para se utilizar os processos de implantação de QoS.

Na FIGURA 4.1, os pontos cinza na base do ambiente de gerência representam os pontos de acesso aos serviços (SAP - *Service Access Point*) [COL 94] existentes e permitem a interação entre o ambiente e os elementos intermediários e bases de dados do modelo. Tais pontos podem ser acessados por qualquer elemento interno do ambiente de gerência (visualização, análise ou implantação de QoS). Entretanto, é pouco esperado que a implantação de QoS utilize os pontos de acesso já que esta é uma atividade que na prática interage pouco com a rede (a não ser quando a ativação de um serviço de QoS implantado mas ainda inativo deve ser realizada).

Como comentado, o ambiente de gerência, por possuir o suporte à tarefa de visualização, é a interface de usuário utilizada pelo administrador da rede. Através do uso do ambiente de gerência por parte do administrador, possivelmente a gerência da rede aconteça de acordo com os seguintes passos:

1) Acesso, via visualização de QoS, dos processos para implantação de QoS. O administrador descreve suas necessidades e sua rede, e o processo de implantação auxilia na escolha da melhor arquitetura de QoS.

2) A efetiva implantação da arquitetura escolhida acontece de forma manual (através da instalação física de equipamentos) e de forma automatizada, onde a implantação de QoS configura os serviços da arquitetura escolhida nos equipamentos já instalados.

3) O administrador solicita uma descoberta de QoS, para que os dispositivos e alvos existentes sejam cadastrados na base de associações. Alternativamente, este processo pode ser executado manualmente, na ausência dos processos de descoberta.

4) A base de associações é complementada com informações fornecidas pelo administrador sobre as associações entre alvos, consumidores de políticas e monitores de QoS.

5) Políticas de operação da rede são definidas e armazenadas na base de políticas. O administrador escolhe em quais alvos as políticas devem ser aplicadas, e o sistema de gerência determina quais são os consumidores de políticas associados a tais alvos.

6) As políticas são transferidas para os consumidores pelo sistema de gerência utilizando o modelo *push* ou *pull*. As políticas são então traduzidas pelos consumidores em ações específicas nos alvos.

7) O gerente da rede escolhe quais as políticas críticas implantadas que devem ser monitoradas, e quais os alvos dessas políticas são de maior interesse. O sistema de gerência determina então quais os monitores de QoS estão associados aos alvos importantes.

8) As políticas são transferidas aos monitores de QoS que passam a verificar o efetivo cumprimento das políticas nos alvos.

9) Uma política que não pôde ser implantada, ou que não está se comportando como esperado faz com que a base *status* seja alterada (pelos consumidores no primeiro caso, e pelos monitores de QoS no segundo caso), e notificações enviadas ao ambiente de gerência. Um sistema de alertas pode então apresentar a notificação conforme o desejo do administrador.

10) A análise de QoS acessa as bases de dados e dispositivos de rede para determinar o comportamento histórico da rede e de seus serviços. O comportamento histórico pode ser consultado pelo gerente e anomalias de comportamento podem gerar notificadas novamente encaminhadas ao gerente.

A seção de exemplos ao final do capítulo apresenta exemplos de como os passos anteriores podem ser executados em dois cenários de gerência de redes diferentes.

4.9 Localização dos elementos

As seções anteriores descreveram cada elemento do modelo proposto e como estes podem ser utilizados em cenários de gerência de redes com QoS. Esta seção irá discutir sobre a localização de tais elementos nas infraestruturas da rede e quantos elementos de um mesmo tipo podem ser utilizados em uma solução.

A localização mais óbvia de um elemento é a localização dos alvos. Estes são localizados dentro dos dispositivos que exercem qualquer papel ativo no fornecimento de QoS. Como já verificado, exemplos de alvos são as interfaces e disciplinas de filas em roteadores. Os processos de marcação, policiamento e conformação de tráfego também são exemplos de alvos. Uma localização menos óbvia de alvos são nos sistemas finais (computadores de usuários da rede). Neste caso, podem existir dois tipos diferentes de alvos: aqueles implementados globalmente ao sistema final, a acessado por várias aplicações; e aqueles alvos implementados internamente nas aplicações. No primeiro caso, o alvo tipicamente é implementado pelo sistema operacional que disponibiliza alguma facilidade de fornecimento de QoS. Um exemplo são as soluções

de marcação de pacotes por processos entre o nível de rede e nível de enlace na pilha de protocolos de uma máquina [GRA 2000d]. No segundo caso, cada aplicação implementa internamente um processo relacionado ao QoS. O exemplo típico são as aplicações com suporte ao RSVP que fazem solicitações de reserva de recursos diretamente. Do ponto de visto do gerenciamento, os alvos implementados globalmente nos sistemas finais são mais adequados porque normalmente possuem uma interface de gerenciamento que possibilita a um processo externo a configuração do alvo. Os alvos implementados diretamente por aplicações têm menos chance de apresentarem algum tipo de interface de gerência, o que inibe a utilização do alvo de interesse no sistema de gerenciamento da rede.

A localização do ambiente do usuário é quase tão óbvia quanto a localização dos alvos. Um ponto central na rede é responsável por executar tarefas de visualização, análise e implantação de QoS e apresentar uma interface gráfica com o usuário do sistema (administrador da rede). A interface do usuário pode ser implementada de diversas formas, e com a atual tendência em gerenciamento baseado na Web, a utilização de tecnologias Web para a criação da interface gráfica de gerência apresenta-se como uma solução bem apropriada. Entretanto, soluções não baseadas na Web podem ser utilizadas, e tem sido a regra de mercado (com algumas exceções). O uso de um único ponto central para o ambiente poderia gerar, em redes maiores, muito tráfego de gerenciamento nas proximidades do ambiente. O uso de um ambiente de usuário distribuído pode ser aplicado, visto que as bases de dados são independentes do ambiente do usuário, e que vários ambientes diferentes podem acessar a mesma base.

As bases de dados podem estar localizadas no mesmo dispositivo que implementa o ambiente de gerência ou então em dispositivos separados. Visto que existem três bases de dados diferentes, algumas podem estar localizadas junto ao ambiente de gerência e outras separadamente. Quando as bases encontram-se junto com o ambiente de gerência, não existe a geração de tráfego na rede que corresponda às comunicações entre ambiente e bases, já que as comunicações são internas. Porém, as bases podem se localizar em dispositivos diferentes àquele que contém o ambiente de gerência e neste caso existiria tráfego na interação entre ambiente e bases.

Apesar da FIGURA 4.1 apresentar apenas uma cópia de cada base, por questões de segurança, várias cópias de uma mesma base poderiam ser utilizadas juntamente com mecanismos de replicação. Várias cópias de uma mesma base também facilitariam a distribuição do tráfego de gerenciamento, gerado pelos monitores de QoS, consumidores de políticas e identificadores de alvos.

A localização dos elementos intermediários na rede é mais complexa. Primeiramente, visto que tais elementos são independentes, eles podem estar localizados em locais diferentes. Os monitores de QoS estão muito atrelados aos alvos relacionados. Assim, é esperado que os monitores estejam localizados dentro do mesmo dispositivo que contém os alvos monitorados. Entretanto, dependendo da implementação dos monitores, eles podem estar localizados próximos aos dispositivos, mas não internamente. Por exemplo, um monitor criado para verificar a banda utilizada em uma interface de um roteador poderia acessar o grupo interface da MIB-II e perceber se uma interface está sendo superutilizada.

Os consumidores de políticas estão, freqüentemente, localizados fora dos dispositivos, mas é esperado que equipamentos mais modernos implementem consumidores de políticas internamente a tais dispositivos. Além disso, consumidores

de políticas podem estar localizados junto ao ambiente do usuário de forma a tornar mais rápida a comunicação entre estes elementos. Neste último caso, um consumidor de políticas funcionaria quase como um *driver* para acesso aos dispositivos e correspondentes alvos.

Por fim, os identificadores de alvos freqüentemente estão localizados junto ao ambiente do usuário, atuando como módulos especiais que procuram na rede por dispositivos com suporte a QoS. Por outro lado, os identificadores de alvos podem também estar localizados em segmentos de rede diferentes do segmento onde se localiza o ambiente de gerência. Neste caso os identificadores funcionam como monitores de segmentos, verificando por tráfego de rede gerado por dispositivos com suporte a QoS (no caso de operarem em modo não intrusivo). O local menos apropriado para um identificador de alvo é dentro dos dispositivos de rede, visto que tais dispositivos e alvos internos são os objetos da procura dos identificadores. Uma exceção seria quando os dispositivos tivessem capacidade de anunciar suas características à rede, registrando-se nas bases de dados. Entretanto, dispositivos com tais características são menos freqüentes, além de não existir uma forma padronizada de proceder com um auto-registro no ambiente de gerência.

TABELA 4.1 – Localização dos elementos do modelo

	Dispositivos	**Proxies**	**Hosts**	**Estação de gerenciamento**
Alvos	x	--	x	Apenas se o alvo tiver papel ativo no fornecimento de QoS
Monitores de QoS	x	x	x	x
Consumidores de políticas	x	x	x	x
Identificadores de alvos	--	x	--	x
Ambiente do usuário	--	--	--	x
Bases de dados	--	--	--	x

A TABELA 4.1 apresenta os elementos do modelo proposto e a possível localização destes elementos nos dispositivos de rede. As células marcadas com um X indicam que o elemento da linha pode estar presente no dispositivo identificado pela coluna. A coluna “dispositivos” denota os equipamentos de rede (roteadores, *switches*, pontes, etc.). “Proxies” são equipamentos usados para suportar elementos ativos que atuam em outros equipamentos (por exemplo, um monitor de QoS localizado dentro de um micro usado para monitorar uma rede). “Hosts” são listados para explicitamente definir elementos localizados e atuantes nos sistemas finais. Por fim, “estação de

gerenciamento" é usado para denotar onde o ambiente de gerência e as bases de dados estão localizados.

4.10 Protocolos do modelo

Esta seção discute sobre as possibilidades de protocolos a serem utilizados em uma implementação do modelo proposto. Os protocolos são sugeridos genericamente, e exemplos são apresentados. Como regra geral, procura-se utilizar protocolos abertos e padronizados nas comunicações. Apesar de alguns protocolos padrão poderem, eventualmente, não serem os mais adequados para tarefas mais críticas, esta escolha permite que uma implementação do modelo seja mais aberta de forma a propiciar maiores facilidades na implementação de novos módulos.

Protocolos dos alvos

Os protocolos utilizados para se realizar a comunicação com os alvos são, na verdade, definidos pelos dispositivos que contêm os alvos. Estes protocolos são, então, dependentes dos fornecedores dos dispositivos, que podem optar por utilizar protocolos padrão ou implementar seus protocolos proprietários. Os protocolos mais comuns encontrados nos alvos são Telnet e SNMP, mas dispositivos mais modernos utilizam HTTP para gerenciamento de configuração, e espera-se que brevemente um número maior de equipamentos suporte também o protocolo COPS. Felizmente os protocolos proprietários são mais raros de serem encontrados.

Apesar da diversidade de possibilidades, os protocolos dos alvos não são uma questão crítica, visto que os elementos intermediários são responsáveis por executar a tradução de protocolos. Assim, quando dois roteadores diferentes, que usam protocolos diferentes, devem ser programados para priorizar um fluxo definido, a tarefa de programação percebe os roteadores diferentes como sendo iguais, devido à tradução de protocolos executada pelos consumidores de políticas associados. Traduções similares também ocorrem nos monitores de QoS e identificadores de alvos.

Protocolos dos elementos intermediários

Os protocolos utilizados para a comunicação com os elementos intermediários devem ser adequados para a realização de algumas operações. Os requisitos para os protocolos de comunicação com os consumidores de políticas são:

1) Permitir a transferência de políticas e notificações do ambiente de gerência para um consumidor de políticas;
2) Permitir a transferência de políticas entre base de políticas e consumidores de políticas;
3) Permitir a notificação do ambiente de gerência sobre problemas decorrentes da implantação de políticas; e
4) Permitir a notificação da base *status* sobre a implantação de uma política aplicada em um alvo.

As necessidades de notificações apontam principalmente para a utilização de alguma versão do SNMP. Como COPS também possui serviços de notificação, o mesmo também poderia ser utilizado. A transferência de políticas, cujas mensagens são maiores que as mensagens geradas pelas notificações, pode ser realizada por algum protocolo de transferência de arquivos. A opção mais óbvia nesse caso é a utilização de FTP ou HTTP, mas o acesso via LDAP também é uma opção interessante, principalmente se a base de políticas for implementada através de um serviço de diretórios. Este último caso é a tendência apontada pelo IETF, mas esta opção ainda encontra-se em processo de padronização no IETF.

A comunicação com os monitores de QoS apresenta os mesmos requisitos que a comunicação com os consumidores de políticas. Como tal, os mesmos exemplos de protocolos anteriormente citados podem ser utilizados para implementar a comunicação dos monitores de QoS com os outros elementos do modelo (ambiente de gerência e bases de dados).

Os requisitos para a comunicação com os identificadores de alvos são:

1) Permitir a transferência de regras de descoberta do ambiente de gerência para os identificadores de alvos;
2) Permitir a transferência, do ambiente de gerência para os identificadores de alvos, de um grupo de endereços de rede que já são conhecidos, mas necessitam ainda da identificação de suas capacidades em relação aos serviços de fornecimento de QoS;
3) Permitir a transferência, dos identificadores de alvos para a base de associações, do conjunto de alvos descobertos, suas identificações e associações com os dispositivos que contenham tais alvos.
4) Notificar o ambiente de gerência quando um bloco de alvos for descoberto, sendo que um bloco pode ser composto por um ou mais alvos.

As notificações enviadas dos identificadores de alvos para o ambiente de gerência tendem a ser maiores que as notificações geradas pelos consumidores de políticas e monitores de QoS. As notificações dos identificadores de alvos devem descrever blocos de alvos encontrados, e o tamanho de cada bloco pode variar de poucos bytes (por exemplo, se o bloco for constituído de apenas um alvo) até muitos bytes (se o bloco contiver vários alvos). Logo, os mecanismos de notificação do SNMP (ou mecanismos similares como COPS) são menos adequados no caso dos identificadores de alvos. Requisições HTTP utilizando mensagens POST poderiam ser utilizadas. Um identificador de alvos mandaria uma mensagem POST ao ambiente de gerência (que se comportaria como um servidor Web nesta operação) e receberia como resposta uma pequena página de confirmação.

Uma solução ainda possível para o acesso aos elementos intermediários é a utilização de Script MIB [QUI 99]. Tal solução visa a implementação de facilidades para a criação de um ambiente de gerência que opere utilizando a delegação de tarefas a gerentes de rede intermediários (MLM – *Mid-Level Managers*) [GOL 96]. Como os elementos intermediários do modelo proposto acabam na realidade desempenhando tarefas de gerentes intermediários, a utilização de Script MIB é uma alternativa consistente e interessante. No capítulo 5 será apresentado um exemplo onde Script MIB é utilizada em uma implementação do modelo.

Protocolos das bases de dados

Indiretamente, alguns protocolos para acesso às bases de dados foram citados nos itens anteriores. Neste item, o uso específico de alguns destes protocolos é sugerido e justificado.

Os protocolos utilizados para acessar as informações das bases de dados são diferentes entre si porque a natureza de cada base é também diferente. A base de políticas deveria ser acessada usando-se um serviço de diretórios como o LDAP [STR 2001] (FIGURA 4.1, setas 2, 7 e 12). Visto que as políticas são informações que possuem pouca atualização, mas podem ser acessadas diversas vezes, um protocolo de poucas escritas e muitas leituras como o LDAP é mais adequado.

As informações das bases de *status* e associações são mais dinâmicas, e o protocolo LDAP, neste caso, deveria ser evitado. Uma implementação possível poderia utiliza as mensagens *InformRequest* do SNMPv2 [CAS 96] para atualizar as informações de estado (setas 4 e 8). As mensagens *InformRequest*, de acordo com as definições do IETF, podem ser encaradas como *traps* SNMP com confirmações. Assim, os monitores de QoS, percebendo uma degradação, podem atualizar a informação de estado de um fluxo ou agregado monitorado notificando a base de *status*. Os consumidores de políticas podem também notificar a base *status* quando a implantação de uma política falhar.

A interação com a base de associações acaba tornando-se uma questão mais complicada quando a base implementada não utiliza um protocolo padrão, e isso tende a acontecer porque as associações não podem ser implementadas, por exemplo, utilizando um serviço de diretórios porque os dados da base freqüentemente sofrem alterações. Isso faz com que a base de associações acabe sendo implementada através de bancos de dados de mercado (por exemplo, Oracle [LON 2000] e Sybase [PAN 96]) que não possuem interfaces de comunicação padronizadas. Por fim, o ambiente de gerência e os identificadores de alvos poderiam também ter acesso à base de associações através de HTTP combinado com o uso de um mecanismos de scripts como o PHP4 [PHP 2001] (setas 11 e 14). Isso abstrairia a implementação da base, fornecendo uma interface única de comunicação.

A TABELA 4.2 resume os possíveis protocolos apresentados utilizados na comunicação entre os elementos do modelo. Células marcadas com um X denotam que o protocolo da linha pode ser utilizado na comunicação com o elemento da coluna.

TABELA 4.2 – Protocolos e elementos do modelo

	Alvos	Consu-midores	Monitores	Identifi-cadores	Asso-ciações	Políticas	Status
SNMP	x	x	x	x	--	--	x
HTTP	x	x	x	x	--	x	--
Telnet	x	x	x	x	--	--	--
COPS	--	x	--	--	--	--	--
LDAP	--	--	--	--	x	--	--

É importante notar que uma implementação do modelo poderia utilizar outros protocolos além dos protocolos discutidos. Entretanto, espera-se que implementações diversas utilizem protocolos que funcionalmente sejam similares aos protocolos sugeridos nesta seção. Por exemplo, na gerência de uma rede ATM, o conjunto de protocolos utilizados pode ser diferente do conjunto anteriormente discutido. Nem por isso, uma rede ATM não poderá implementar uma solução baseada no modelo. A diferença, entretanto, é que a solução ATM utilizará alguns protocolos diferentes, mas funcionalmente similares àqueles aqui sugeridos.

4.11 Exemplos

Esta seção apresenta alguns cenários de gerenciamento de QoS com o objetivo de mostrar como ocorrem as interações entre os elementos do modelo proposto. Dois exemplos são apresentados: priorização de tráfego em uma rede de supermercados e ajustes na utilização dos recursos de rede de uma corporação.

Manutenção da rede de um supermercado automatizado

Neste primeiro exemplo será possível verificar como a utilização do modelo proposto pode ser empregada na priorização de tráfego de aplicações de missão crítica. Tais aplicações exigem priorizações sobre as demais aplicações, mas normalmente não exigem garantias estritas em relação ao atraso e *jitter*. Tipicamente são aplicações convencionais (por exemplo, FTP, navegação Web, e-mail, etc.) que em um contexto específico necessitam um nível de garantia, em relação às outras aplicações, maior que o normalmente fornecido pela rede.

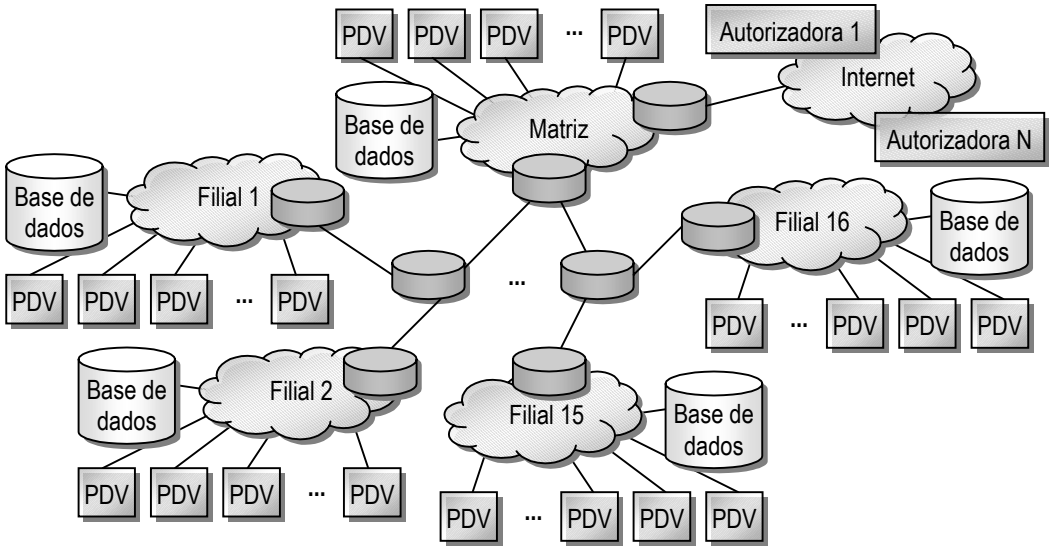

FIGURA 4.2 – Rede hipotética de uma cadeia de supermercados

No cenário específico deste exemplo, imagina-se uma rede de computadores interligando uma matriz a 16 filiais de uma cadeia de supermercados. O ponto de acesso da corporação à Internet é único e centralizado na matriz. O tráfego Web é intenso entre

as filiais e a matriz quando for necessário o acesso às autorizadoras de créditos (bancos ou operadoras de cartões) localizadas fora da rede interna (acessíveis via Internet). A FIGURA 4.2 apresenta um esquema geral da rede.

Cada filial possui sua base de dados local que armazena o movimento realizado pelos pontos de venda (PDV). Arquivos gerados nos PDVs descrevem as vendas realizadas e devem ser enviados à base de dados local da filial via FTP. O tráfego de rede durante o dia é intenso localmente às filiais para a atualização das bases locais, e entre filiais e matriz quando um PDV precisa autorizar, na Internet, um pagamento de compra realizado com cartão bancário ou de crédito.

Entretanto, entre 23h e 1h, duas operações críticas acontecem: deve-se transferir os dados das vendas acumulados durante o dia das bases de dados locais das filias para o servidor corporativo; e todas as autorizadoras de crédito enviam arquivos descritivos das operações em cartão (bancário ou crédito) realizadas durante o dia para confirmação junto à cadeia de supermercados.

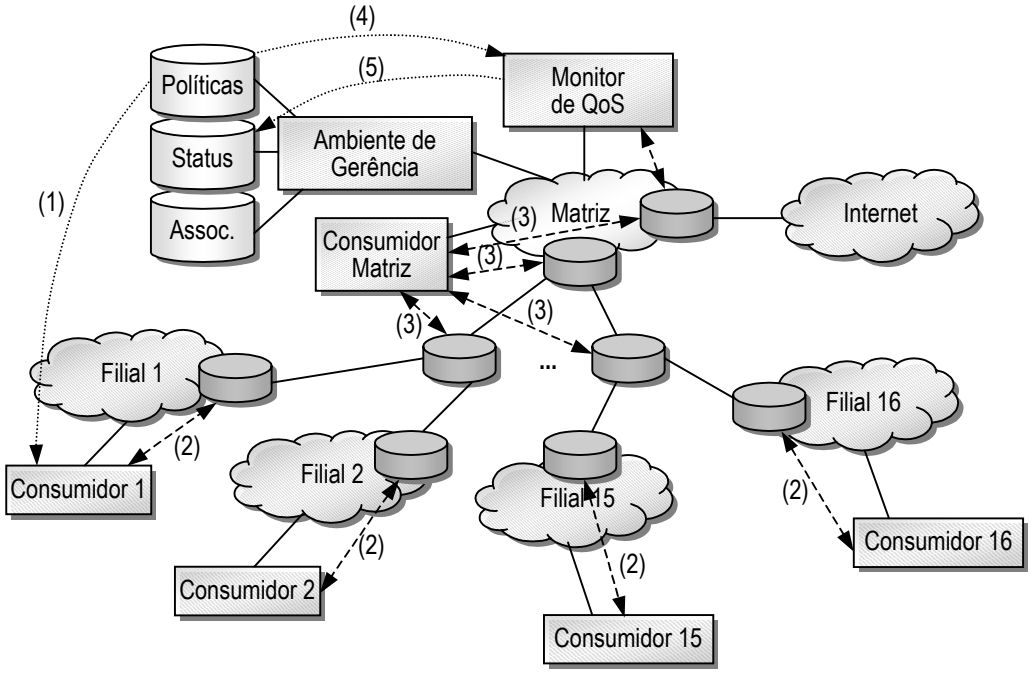

FIGURA 4.3 – Componentes do modelo aplicados a cadeia de supermercados

Neste cenário, o administrador da rede deve garantir que o seguinte conjunto de operações seja realizado adequadamente:

1) Transferências de arquivos descritivos de venda dos PDVs para a base de dados local da filial (via FTP), das 8 horas às 23 horas de segunda a sábado;
2) Comunicação para autorização de crédito entre PDVs e autorizadoras acessíveis via Internet (utilizando protocolo proprietário de cada autorizadora), das 8 horas às 23 horas de segunda a sábado;
3) Transferências das bases de dados das filiais à matriz (via método POST do HTTP), das 24 horas à 1 hora de segunda a sábado;
4) Transferências de arquivos de conferência de transações de autorização de crédito das autorizadoras à matriz (via SSH2 [BAR 2001]), das 23 às 24 horas de segunda a domingo.

Imaginando que o administrador de rede possui uma implementação do modelo de gerência de QoS proposto, inicialmente a rede existente deve ser mapeada. Os dispositivos encontrados e cadastrados são então repassados à descoberta de QoS que irá identificar quais os mecanismos relacionados ao fornecimento de QoS que estão presentes nos equipamentos. Imaginando que a rede não possa ser alterada, a implantação de QoS procede com a ativação dos serviços com QoS descobertos e a descoberta registra os serviços existentes, alvos e dispositivos na base de associações. O administrador deve então instalar consumidores de políticas e monitores de QoS na rede para que seja gerenciada (assume-se neste exemplo que os identificadores de alvos estão localizados junto ao ambiente de gerência). O administrador então registra na base de associações os consumidores, monitores e identificadores de alvos instalados na rede. Após a implantação, a rede apresentará os componentes mostrados na FIGURA 4.3.

Para o adequado funcionamento das operações anteriormente listadas, o administrador da rede cria 4 políticas, apresentadas na FIGURA 4.4. O sistema de gerência deve possuir facilidades para a criação de políticas utilizando uma linguagem que permita a descrição das mesmas. Não existe ainda uma linguagem de definição de políticas padrão. Algumas propostas recentes existem (como por exemplo, a linguagem Ponder [DAM 2001]), mas as pesquisas em relação a esta questão devem desenvolver ainda diversos trabalhos. No exemplo da FIGURA 4.4 utilizou-se uma linguagem hipotética, no mesmo estilo dos exemplos de políticas existentes nos documentos do IETF.

PDV2DB
```
if ((net_source =net_dest)
  and (app = FTP)
  and (8h ≤ time ≤ 23h)
  and (mon ≤ dayw ≤ sat))
then
  priority ← high
endif
```

PDV2INTERNET
```
if ((net_dest = Internet)
  and ((app = visa)
    or (app = mastercard)
    or (app = bb))
  and (8h ≤ time ≤ 23h)
  and (mon ≤ dayw ≤ sat))
then
  priority ← high
endif
```

DB2SERVER
```
if ((host_dest = server)
  and (app = HTTP)
  and (0h ≤ time ≤ 1h)
  and (mon ≤ dayw ≤ sat))
then
  priority ← high
endif
```

INTERNET2SERVER
```
if ((host_dest = server)
  and (app = SSH2)
  and (23h ≤ time ≤ 24h))
then
  priority ← high
endif
```

net = endereço da rede origem ou destino;
net_source = endereço da rede origem;
net_dest = endereço da rede destino;
internet = endereços externos à rede gerenciada;
mon-sun = dias da seman;

app = porta da aplicação origem ou destino;
app_source = porta da aplicação origem;
app_dest = porta da aplicação destino;
time = hora do dia;
dayw = dia da semana;

FIGURA 4.4 – Políticas para a cadeia de supermercados

Cada política é identificada por um nome único no sistema (no caso do exemplo, as políticas são: PDV2BD, PDV2INTERNET, BD2SERVER e INTERNET2SERVER). Por convenção do exemplo, o primeiro nome indica quem inicia a transmissão e o segundo (após o 2) indica quem recebe uma transmissão. Cada política pode possuir diversas regras (*policy rules*) no formato IF-THEN. Políticas podem também ser formadas pela combinação de outras políticas, permitindo que políticas complexas sejam formadas a partir de combinação de diversas políticas simples. As políticas definidas para a cadeia de supermercados possuíam internamente uma regra apenas. Cada regra, por sua vez, é

formada por um conjunto de condições (IF) e um conjunto de ações (THEN). Por exemplo, a política PDV2BD procede apenas se os endereços de origem e destino de um pacote forem de máquinas de uma mesma sub-rede. Além disso, a aplicação associada deve ser FTP (porta de origem ou destino iguais a 21, para controle, ou 20, para dados). Por fim, a política só é valida entre 8 horas e 23 horas, de segunda a sábado. Quando todas as condições da política forem verdadeiras, a ação é aplicada ao pacote, que no caso é o fornecimento de alta prioridade de encaminhamento. As demais políticas são definidas de forma semelhante, e o resultado aplicado quando as condições das regras forem verdadeiras é sempre o de atribuir alta prioridade aos pacotes.

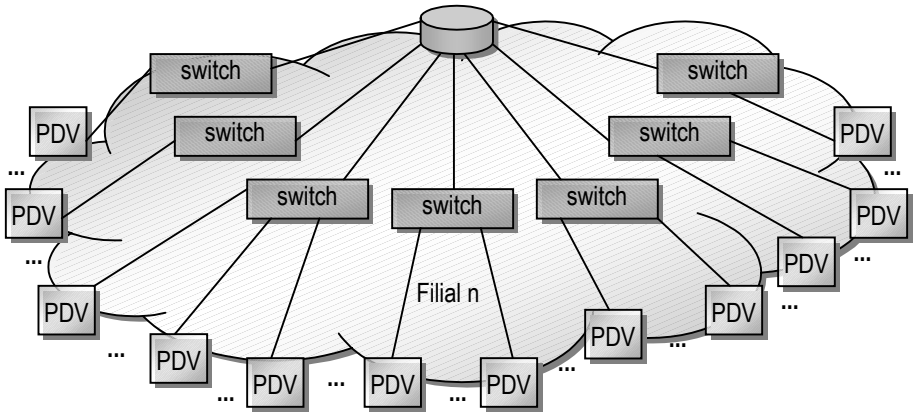

FIGURA 4.5 – Rede interna de uma filial da cadeia de supermercados

Assim que definidas, as políticas são então armazenada na base de políticas e o administrador define em quais alvos tais políticas devem ser implantadas. Em relação aos dispositivos e alvos, o exemplo possui os seguintes tipos de roteadores na rede:

- Assumindo que cada rede local (filial) é baseada em *switches* e que existe internamente largura de banda suficiente por porta de roteador, os roteadores de cada filial (um por filial) possuem 8 portas Ethernet 10 Mbps. Destas, 7 portas são utilizadas para interligação com *switches* (FIGURA 4.5), e uma é utilizada para ligação com a rede externa. Cada roteador deste tipo possui suporte a priorização via serviços integrados, baseada em porta de aplicação origem e destino, endereço de rede origem e destino, e protocolo de transporte (TCP ou UDP). Cada interface possui quatro filas com prioridades alta, média, normal (*best-effort*) e baixa, além de um processo de classificação de pacotes (que determina através de qual fila cada pacote deve ser encaminhado). O acesso aos equipamentos é exclusivamente via Telnet. Como cada roteador possui 8 interfaces, 1 classificador por interface e 4 filas por interface, existirão 40 alvos (5 x 8) por roteador de filial.

- Roteadores intermediários são aqueles que conectam grupos de filiais à matriz. Cada roteador intermediário é ligado a dois outros roteadores de filiais, e ainda interligado ao roteador da matriz. Assim, existirão 3 interfaces ativas em cada roteador deste tipo. Internamente, cada roteador possui facilidades para suporte a serviços diferenciados, três filas de prioridades diferentes (*expedited* [JAC 99], *assured* [HEI 99] e *best-effort*) e um processo de classificação de pacotes global ao roteador. O acesso e configuração dos mesmos são realizados via SNMP, e internamente existe

ainda suporte à MIB RMON2. Todas as interfaces são também de 10 Mbps. Em cada roteador intermediário são encontrados então 10 alvos: 3 (interfaces) x 3 (filas) + 1 (classificador).

- O roteador da matriz é similar aos roteadores intermediários, com a única diferença de possuir 9 interfaces ativas: 8 utilizadas para a comunicação com os roteadores intermediários e 1 para comunicação com a rede interna. Neste caso, o número de alvos existente será de 28: 9 (interfaces) x 3 (filas) + 1 (classificador).

- Por fim, o roteador de acesso à Internet possui apenas 2 interfaces: 1 interface Ethernet 10 Mbps para ligação com a rede local e 1 interface WAN 128 Kbps para ligação com a Internet. O roteador suporta também serviços diferenciados com 3 filas e um processo de classificação global. Logo, existirão 7 alvos: 2 (interfaces) x 3 (filas) + 1 (classificador).

O número total de alvos de todos os roteadores de filiais é de 640: 16 (roteadores de filiais) x 40 (alvos por roteador). O número de alvos de roteadores intermediários é de 80: 8 (roteadores intermediários) x 10 (alvos por roteador). Somando com os número de alvos do roteador da matriz (28 alvos) mais os alvos do roteador para acesso à Internet (7 alvos), tem-se na rede um total de 755 alvos!

A próxima operação a ser realizada pelo administrador é a transferência das políticas definidas anteriormente para os consumidores adequados (FIGURA 4.3, seta 1). Os alvos, no processo de descoberta, foram classificados e no caso específico do exemplo os seguintes tipos de alvos são encontrados na rede: classificadores de pacotes e gerenciadores de disciplinas de filas. Espera-se que o sistema de gerência apresente uma interface de usuário que facilite a determinação dos alvos adequados. A TABELA 4.3apresenta a correspondência entre políticas e os alvos onde elas devem atuar.

A implantação de uma política em um alvo gera uma atualização na base de associações (para "ligar" as políticas aos alvos) e uma nova entrada na base de *status* é criada para identificar o estado desse relacionamento. O estado da política associada ao alvo é inicializado com um valor que indique algo como "em transferência" (por exemplo, valor 0). O sistema de gerência determina então, acessando a base de associações, quem são os consumidores de políticas associados aos alvos e notifica cada um da existência de novas políticas. Nesta solução, estaria sendo utilizado o modelo *pull* para transferência de políticas aos consumidores.

Os consumidores notificados acessam a base de políticas e recuperam as mesmas. De posse de todas as políticas adequadas, o consumidor de políticas pode então analisar a existência de conflitos. Quando duas políticas encontrarem-se em conflito dentro de um mesmo consumidor, algum mecanismo de resolução deve ser utilizado, por exemplo, o uso de prioridades entre políticas [LUP 99]. Nestas situações, a base de *status* é notificada para indicar aquelas políticas que não puderam ser implantadas em decorrência de conflitos. A verificação de conflitos também pode ser executada pelo sistema de gerência no momento que uma política é criada ou transferida para o repositório. Assumindo que não foram encontrados conflitos entres as políticas no ambiente de gerência, no repositório e no consumidor, então notificações são enviadas dos consumidores à base *status* indicando que cada política está agora no estado "em implantação" (valor 1). Os consumidores acessam então os alvos adequados e configuram os mesmos para que as políticas sejam aplicadas. Entretanto, políticas que

devem ser ativadas no futuro, por exemplo, tem seu estado alterado para "suspensa" (valor 2) na base *status*. As políticas transferidas aos consumidores são apenas ativadas nos momentos definidos. Quando isso acontece, as políticas são então traduzidas e o consumidor programa os alvos. Por exemplo, a política DB2SERVER só é ativada à meio-noite. Os roteadores são programados para fornecer a mais alta prioridade existente aos fluxos que se dirigem ao servidor da corporação. Os consumidores que atuam nos roteadores das filiais abrem sessões Telnet (setas 2) e programam a classificação de pacotes nas interfaces de entrada, e programam as filas da interface de saída (em direção à matriz). O consumidor associado aos roteadores intermediários e ao roteador da matriz programam, via SNMP (setas 3), a classificação dos pacotes de acordo com as definições de serviços integrados para que tais pacotes sejam atendidos pela fila Gold (de mais alta prioridade).

TABELA 4.3 – Correspondência entre alvos e políticas

Alvos / Políticas		PDV2DB	PDV2 INTERNET	DB2 SERVER	INTERNET2 SERVER
Roteadores de filiais	Classificadores de pacotes das interfaces internas à filial	X	X	X	
	Gerenciadores de filas das interfaces internas à filial	X	X	X	
	Classificadores de pacotes da interface para a rede externa		X	X	
	Gerenciador de filas da interface para a rede externa		X	X	
Roteadores intermediários	Classificador de pacotes para todas as interfaces		X	X	
	Gerenciadores de filas das interfaces para as filiais		X	X	
	Gerenciador de filas da interface para a matriz		X	X	
Roteador da matriz	Classificadores de pacotes para todas as interfaces			X	
	Gerenciadores de filas das interfaces para as filiais		X	X	
	Gerenciador de filas da interface para a rede interna da Matriz		X	X	
Roteador de acesso à Internet	Classificadores de pacotes para todas as interfaces				X
	Gerenciador de filas da interface para a rede da matriz		X		X
	Gerenciador de filas da interface para a Internet		X		X

Para os fluxos mais críticos (por exemplo, a transferência de arquivos descritivos de operações de débito/crédito), o administrador da rede deve definir um processo de

monitoração. Como os fluxos são descritos a partir das políticas, o administrador determina qual a política crítica que deve ser monitoração (INTERNET2SERVER, por exemplo) e o sistema de gerência encarrega-se de verificar a rede. Para isso, os alvos associados às políticas são determinados com uma consulta a base de associações. Para cada alvo são apontados os monitores de QoS associados (quando existirem). Na rede exemplo, um único monitor de QoS é utilizado para monitorar todos os roteadores. Os monitores identificados são então contatados para que a política a ser monitorada seja transferida (seta 4). O QoS definido para todas as políticas do exemplo definia apenas a prioridade do fluxos. Assim, o monitor de QoS deve verificar junto ao roteador de acesso à Internet se todos os pacotes SSH2 destinados ao servidor, entre 23h e 24h estão sendo priorizados em relação aos outros pacotes. Isso pode ser conseguido no exemplo porque o roteador de acesso à Internet possui uma MIB RMON2 que permite identificar a vazão associada a cada fluxo de dados. Se os fluxos estiverem sendo inadequadamente priorizados, então a estação de gerência é notificada sobre o evento e a base *status* tem o valor associado à política alterado (seta 5) para o valor "em congestionamento" (valor 3).

Para a análise mais adequada do comportamento da rede em relação às priorizações, o administrador procede então com a tarefa de análise de QoS. O ambiente de gerência passa a coletar continuamente informações nos roteadores durante o período em que as políticas estão ativas. Novamente, como muitos roteadores possuem a MIB RMON2, e a preocupação principal é sobre a vazão obtida por cada fluxo, o processo de análise pode fornecer dados adequados. O administrador, de posse dos dados resultantes pode decidir, por exemplo, diminuir a prioridade de fluxos atribuída em uma política se os recursos existentes forem suficientes.

Por fim, o administrador pode ainda solicitar aos processos de descoberta de QoS um conjunto de operações periódicas, que podem ser utilizadas para registrar quais os PDVs que estão ativos. Novos PDVs anteriormente não cadastrados podem ser encontrados, e uma filial interna recentemente implantada pode ser totalmente mapeada. A análise de QoS pode ser aqui combinada com a descoberta. Imaginando que um processo inteiro de descoberta apresente uma espécie de "fotografia" da rede gerenciada, a comparação entre diversas "fotografias" pode revelar os períodos mais críticos de utilização da rede, os PDVs mais freqüentemente em operação, etc. A análise de QoS seria a tarefa responsável por fazer tais comparações das "fotografias" geradas pela descoberta de QoS.

Videoconferência para o diretor do departamento de vendas da rede congestionada

Neste segundo exemplo de utilização do modelo de gerência de QoS proposto, serão abordados aspectos complementares que podem ser suportados nas implementações do modelo. Em particular, o nível de abstração de informações conseguido em cada implementação é discutido. Além disso, neste exemplo, assume-se que os requisitos de QoS exigidos pelos os usuários não se restringem apenas a banda disponível, mas também estão relacionados com atraso, perda e *jitter*. Estes requisitos são decorrentes da utilização de aplicações com restrições temporais severas que não toleram serviços pouco precisos. Logo, o presente exemplo acaba por complementar o exemplo anterior por abordar um conjunto diferente de aspectos.

No exemplo anterior do uso de um sistema de gerência que implementava o

modelo proposto, as políticas eram definidas a partir de uma linguagem de alto nível e traduzidas, nos consumidores de políticas, por ações específicas nos dispositivos. Abstrações ainda maiores podem ser utilizadas na representação de políticas, na tentativa de "esconder" ainda mais complexidades do administrador de rede. O nível de abstração conseguido depende exclusivamente dos mecanismos de tradução existentes, e os consumidores de políticas e monitores de QoS implementam uma primeira abstração básica importante. Outras abstrações acabam sendo implementadas dentro do sistema de gerência e estão relacionadas não apenas com as políticas, mas também com alvos e outros elementos que compõem o sistema. Nesta seção, é apresentado um exemplo de utilização do modelo onde os níveis de abstração supostamente fornecidos pela implementação do sistema são maiores em comparação com aqueles encontrados no exemplo anterior.

Supõe-se neste exemplo uma rede de uma corporação onde o diretor do departamento de vendas realiza reuniões com os clientes preferenciais através de sessões de videoconferência. A rede corporativa, entretanto, vem enfrentando diversos problemas de congestionamento porque muitos usuários começaram a utilizar um *software* que apresenta conteúdo multimídia no monitor quando o usuário permanece um período de tempo sem utilizar seu computador. O problema principal é que o conteúdo multimídia é recuperado da Internet, o que gera congestionamentos que prejudicam, por exemplo, o setor financeiro que utiliza *software* de *homebacking* para pagamento dos funcionários. A inclusão da videoconferência solicitada pelo diretor do departamento de vendas parece ser utópica, já que a rede encontra-se em péssimo estado.

Neste contexto, o administrador tem dois desafios: reduzir a interferência do tráfego gerado pela nova aplicação multimídia, e fornecer recursos de rede suficientes para que o diretor da área de vendas consiga realizar suas sessões de videoconferência com os clientes prioritários. Utilizando uma linguagem para criação de políticas com alto grau de abstração, os objetivos da rede poderiam ser definidos através das política apresentada na FIGURA 4.6.

BUSINESS GOAL

Rule1	**Rule2**

if ((trafficToOrFrom salesDirectorHost)
and (trafficType is videoConference))
then
trafficQoSClass ← videoConferenceClass
endif
	if (trafficToOrFrom Internet)
	and (trafficType is badScreenSaverApp))
	then
	trafficQoSClass ← worstThanBestEffort
	endif

trafficToOrFrom = tráfego destinado ou gerador por;
trafficType = tipo de tráfego;
trafficQoSClass = classe de QoS a ser aplicada;
videoConferenceClass = QoS adequado a videoconferência;
worstThanBestEffort = QoS inferior ao *best-effort*;
salesDirectorHost = máquina do diretor de vendas;
Internet = máquinas fora da corporação;
videoConference = tráfego de videoconferência;
badScreenSaverApp = tráfego do *screen saver;*

FIGURA 4.6 – Política para a solução de videoconferência

É importante notar que na definição da política BUSINESS GOAL, foram utilizadas duas regras que cobrem os objetivos principais. Uma outra opção seria a definição de duas políticas separadas, cada uma contendo uma regra da política original. Quando as

regras existentes são separadas em muitas políticas, o conjunto total disponibilizado ao administrador pode se extremamente grande. Logo, a utilização de poucas políticas, mas com um número maior de regras internamente resolve o problema. Por outro lado, como as regras acabam se concentrando, regras que fazem parte de uma mesma política serão aplicadas nos alvos selecionados, muitas vezes, entretanto, sem necessidade. Por exemplo, as regras definidas na FIGURA 4.6 relacionam-se com aspectos diferentes que tentam levar a um objetivo comum: o funcionamento adequado da rede de acordo com os objetivos da corporação. Na primeira regra, todo tráfego de videoconferência originado ou destinado à máquina do diretor recebe um tratamento especial. Na segunda regra, todo tráfego *screensaver* originado ou destinado à Internet recebe uma priorização baixa. A primeira regra deve ser aplicada a todos os roteadores da rede que fazem parte do caminho da máquina do diretor até à Internet. A segunda regra pode ser apenas aplicada no roteador da corporação que se conecta à Internet. Entretanto, como as regras fazem parte de uma mesma política, as regras acabarão sendo utilizadas por todos os roteadores que podem implantar em seus alvos tal política, o que faz com que, neste caso, a segunda regra também seja implantada, mesmo sem necessidade, nos roteadores que fazem parte do caminho entre a máquina do diretor de vendas até a Internet. A decisão de como estruturar as regras em políticas acaba sendo de responsabilidade do administrador, que deve ponderar, de acordo com suas necessidades e seu ambiente gerenciado, qual a melhor estratégia a ser adotada.

Em relação à monitoração de QoS, o tratamento de políticas e regras também é crucial porque o que acaba sendo transferido aos monitores são políticas inteiras, e estas são monitoradas considerando todas as suas regras. Entretanto, provavelmente nem todas as regras de uma política transferida a um monitor de QoS mereçam monitoração. Isso sugere duas opções de tratamento de políticas e regras por uma implementação do modelo: políticas como unidades de operação, ou regras como unidades de operação. Para melhor compreensão destes conceitos, inicialmente devem ficar claros outros dois: unidades de definição, e unidades de operação. As unidades de definição são as porções mínimas utilizadas para expressar como as partes de uma rede devem operar. Sem controvérsia alguma, felizmente, as unidades de definição são as regras das políticas. As unidades de operação são os blocos de definições transferidos aos elementos do modelo para que estes programem/monitorem a rede que deve passar a operar de acordo com as definições. As unidades de operação de um sistema baseado em políticas são as políticas. Entretanto, o modelo apresentado não se restringe ao uso de políticas e como tal, é necessária uma reflexão extra sobre as unidades de operação utilizadas. Utilizando-se políticas como unidade de operação pode-se chegar a situações como as descritas anteriormente, onde porções da rede são monitoradas sem necessidade. Se as unidades de operação forem as regras, tais situações podem ser totalmente controladas, mas o nível de complexidade do sistema de gerência aumenta porque o administrador passa a operar com um conjunto maior de possibilidades. Uma situação intermediária e mais flexível, entretanto, pode ser adotada em uma implementação. Pode-se habilitar as políticas como unidades de controle padrão durante toda a operação da rede. Em situações específicas, onde o administrador achar adequado um controle maior sobre o que é monitorado ou implantado, o uso de regras como unidades de controle passa a ser então utilizado.

Neste segundo exemplo, o sistema de gerência possui um conjunto de classes de QoS que podem ser aplicadas aos tráfegos da rede (foram utilizadas nas regras as classes *videoConferenceClass* e *worstThanBestEffort*). O administrador pode manipular,

utilizando ferramentas adequadas, o conjunto de classes criando, removendo e alterando as classes existentes. Cada classe de QoS define pelo menos quatro parâmetros que devem tentar ser fornecidos ao fluxo que recebe uma determinada classe: vazão, perda, atraso e *jitter*. Uma classe pode ser definida, entretanto, com apenas alguns desses parâmetros, significando que os outros não influenciam tanto. Por exemplo, em um fluxo de controle de videoconferência, a perda de pacotes deve ser evitada. Já nos fluxos de dados (vídeo e voz) da mesma videoconferência, a perda não é tão importante quanto são a vazão e o atraso. Tais classes podem ser armazenadas no repositório de políticas e recuperadas toda vez que uma política for implantada, ou toda vez que um monitor de QoS precisar verificar a efetividade de uma política em andamento.

O sistema de gerência é responsável por realizar neste instante uma tradução (implementando uma abstração) importante: traduzir as classes de QoS em priorizações enviadas aos consumidores de políticas, como no exemplo da cadeia de supermercados. Os consumidores então procedem ainda com uma outra tradução. O sistema de gerência deve também traduzir internamente as expressões utilizadas para identificar as aplicações. A tradução mais simples neste caso é a direta, onde uma expressão dá origem, por exemplo, a um número de porta de aplicação (a expressão *http* seria traduzida para a porta 80). Traduções mais complexas, por outro lado, podem tomar uma expressão e gerar diversas informações relacionadas. Por exemplo, se a expressão *videoConference* relacionar-se com interações H.323 [ITU 98] a tradução irá gerar todo o conjunto de portas necessário para que aplicações H.323 operem adequadamente. Outros elementos que o sistema de gerência deve traduzir são as identificações de redes e máquinas origem ou destino. As máquinas origem e destino podem ser facilmente traduzidas, por exemplo, através de serviços DNS ou similares. As traduções de endereços de rede devem resolver informações no formato (endereço de rede, número de bits para *hosts*). Por exemplo, supondo que a rede local da corporação possui o endereço 200.132.73.0 classe C, o termo *Internet* será traduzido para *not (200.132.73.0/8)*.

```
POLICY DEPLOYMENT

targets = select * from targets where
            addr withinPath (salesDirectorHost, Internet)
              and
            type=priorityEnabled;
targets.deployPolicy (BusinessGoal);
```

FIGURA 4.7 – Implantação de política com linguagem de alto nível

A TABELA 4.3 apresentou um exemplo de um conjunto de correspondências entre alvos e políticas. Uma das tarefas mais complexas, mesmo em um sistema baseado em políticas, é a determinação de quais são os alvos que devem ser utilizados na aplicação de uma política. Um sistema de gerência adequado deve possuir mecanismos que permitam a seleção automática de alvos a partir de consultas do administrador. Novamente, espera-se que uma implementação do modelo seja capaz de facilitar, ou até mesmo automatizar, a escolha dos alvos adequados para a implantação de uma política. A FIGURA 4.7 exemplifica uma possível solução para a implantação das regras definidas na FIGURA 4.6 utilizando uma linguagem onde o gerente define que tipos de alvos devem ser utilizados e o sistema seleciona tais alvos para que o administrador

proceda com a implantação de políticas.

Na primeira expressão são selecionadas todos os atributos (*) de todos os alvos (*targets*) cujos endereços (*addr*) façam parte do caminho (*withinPath*) de interligação da máquina do diretor do setor de vendas com a Internet. Além disso, os alvos devem possuir mecanismos de priorização habilitados (*priorityEnabled*). A segunda expressão permite então que as políticas definidas de acordo com os objetivos da rede (*BusinessGoal*) sejam aplicadas (*deployPolicy*) nos alvos anteriormente selecionados.

```
POLICY MONITORING
BusinessGoal.Monitoring = true;

POLICY RULE #1 MONITORING
BusinessGoal.Rule1.Monitoring = true;

POLICY RULE #2 MONITORING
BusinessGoal.Rule2.Monitoring = true;
```

FIGURA 4.8 – Definição de monitoração de políticas em alto nível

Assim como a implantação de políticas deve ser facilitada, da mesma forma deve ser a monitoração de QoS associada. Uma linguagem de alto nível também poderia ser utilizada para determinar quais os pontos críticos da rede e quais as políticas importantes que mereceriam monitoração. Comparativamente, entretanto, a seleção de quais monitores devem ser utilizados é mais fácil que a determinação de quais os consumidores importantes na implantação de uma política. A complexidade na escolha dos consumidores está diretamente relacionada com a complexidade de escolha dos alvos, como visto anteriormente. Quando a monitoração deve ser aplicada, entretanto, os alvos em que uma política deve ser implantada já foram escolhidos, e podem ser facilmente determinados através de consultas à base de associações. De posse de todos os alvos escolhidos, determinar quais os monitores associados é simples, pois também requer apenas consultas à base de associações. Admitindo que a implementação do sistema de gerência utilizado neste exemplo implemente uma linguagem de gerência de mais alto nível, a FIGURA 4.8 apresenta três diferentes possibilidades para a execução da monitoração de QoS para a política BUSINESS GOAL anteriormente definida. Na primeira opção a política e todas as suas regras são monitoradas em todos os alvos que implementam a política. Na segunda opção, apenas a primeira regra é monitorada, e isso é feito em todos os alvos associados à política. Por fim, na terceira opção apenas a segunda regra é monitorada. Não existe a opção de monitorar apenas alvos específicos neste caso, já que todos os alvos associados a uma política implementam tal política. Refinamentos maiores na linguagem poderiam, entretanto, permitir este nível de especificação. Novamente, isso irá depender da implementação do modelo utilizada.

4.12 Especificação SDL do modelo

Nesta seção é apresentada a especificação formal em SDL [BEL 91] das definições de gerenciamento integrado de QoS, de acordo com o modelo proposto. O SDL é utilizado por possuir uma sintaxe gráfica (além da sintaxe textual). Isso facilita a compreensão da especificação. Além disso, especificações em SDL têm sido utilizadas

para definir diversos trabalhos realizados em redes de computadores, provando também ser um bom mecanismo de documentação.

Inicialmente é apresentada a definição SDL do sistema, que fornece o nível mais alto de abstração da especificação (FIGURA 4.9). Cada bloco do sistema é então descrito seguido de seus processos. Novos blocos são descritos na seqüência, após a descrição dos processos do bloco anterior.

Sistema

O sistema descrito é formado por blocos que se comunicam entre si através de canais de comunicação (FIGURA 4.9). A descrição do sistema não apresenta nenhuma definição sobre o comportamento do modelo, mas mostra a estruturação do mesmo. O comportamento do modelo é descrito nos processos que fazem parte da especificação SDL, apresentados dentro de cada bloco do sistema.

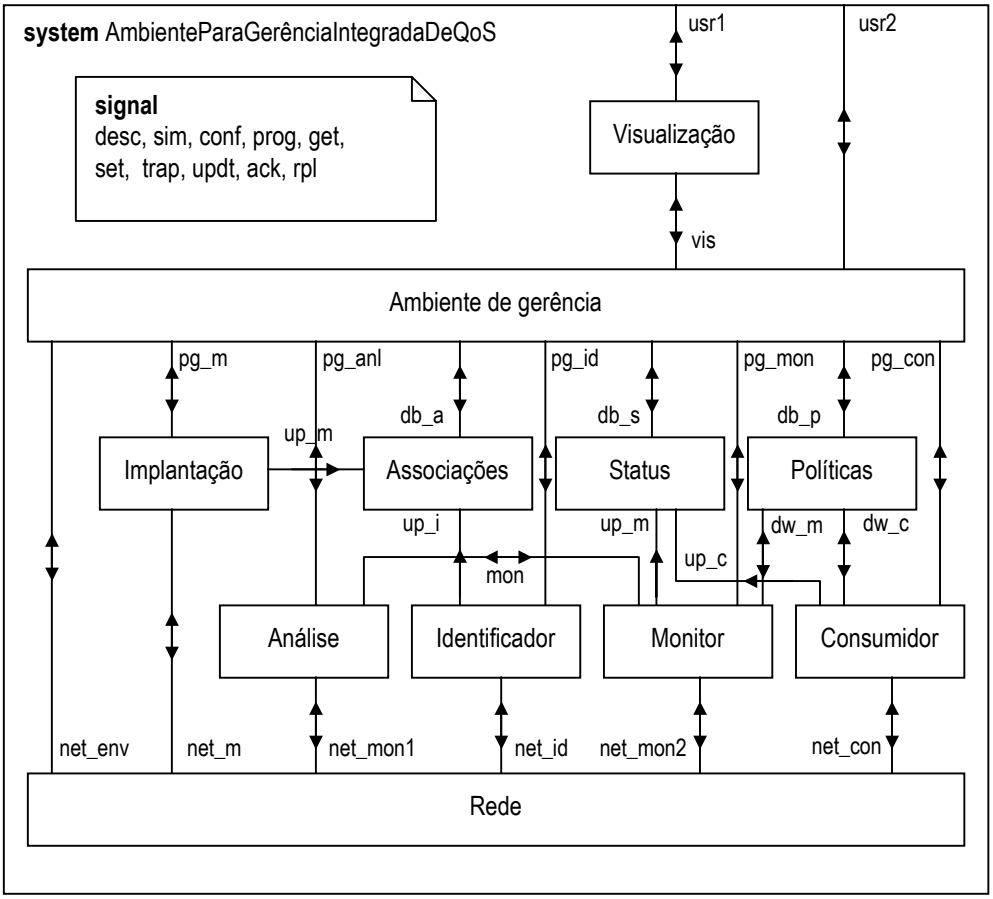

FIGURA 4.9 – Definição SDL do sistema

A rede gerenciada é descrita pelo bloco “Rede” e comunica-se com os outros blocos definidos através dos canais net_env, net_m, net_mon1, net_mon2, net_id e net_com. Cada um destes canais permite a interação bidirecional com o ambiente de gerência, com o bloco para implantação de QoS, com os blocos de análise e monitoração de QoS, com o bloco de identificação de alvos e com o bloco de consumidores de políticas.

Os sinais utilizados dentro dos canais são desc, sim, conf, prog, get, set, trap, updt, ack, rpl e são utilizados nas definições que seguem.

Os blocos “Associações”, “*Status*” e “Políticas” descrevem respectivamente as bases de associações, *status* e o repositório de políticas do modelo. Como a implementação destes blocos conceitualmente é simples, por serem bases de dados, tais blocos são definidos apenas do diagrama do sistema da FIGURA 4.9. O bloco “Rede” também é apenas apresentado no diagrama do sistema, pois representa a rede gerenciada. Como a rede gerenciada não faz parte do ambiente de gerência, mas sim é o alvo maior de tal gerência, os processos internos à rede não são apresentados. Por fim, o bloco de visualização de QoS permite o processamento de dados provindos do ambiente de gerência para que sejam melhor apresentados ao administrador da rede. Como a visualização de QoS é específica de cada implementação, definir o bloco de visualização acabaria por restringir a mesma.

Nas subseções a seguir, os blocos “Implantação”, “Análise”, “Identificador”, “Monitor”, “Consumidor”, “Ambiente de gerência” são formalmente definidos através da apresentação de sua estruturação interna, seus processos, e a comunicação entre os mesmos através de rotas de sinais.

Implantação

O bloco para implantação de QoS (FIGURA 4.10) é composto por três processos: descrição, simulação e configuração.

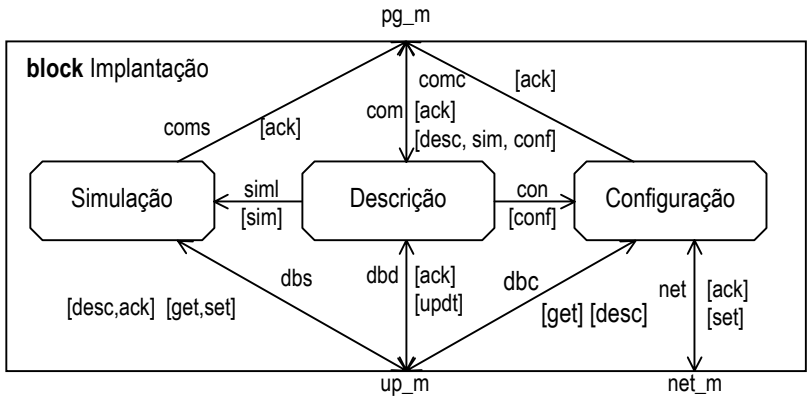

FIGURA 4.10 – Bloco de implantação de QoS

O processo de descrição (FIGURA 4.11) recebe sinais externos (provindos do sistema) que descrevem a rede analisada. O processo de descrição então processa o sinal e gera um sinal enviado a base de associações através da rota de sinais dbd. Esta operação faz com o que base de associações contenha então a rede descrita enviada ao processo de implantação.

Outro sinal de entrada é o sinal de simulação (sim), que é repassado ao processo de simulação (FIGURA 4.12). Neste momento o processo acessa a base de associações,

recupera a descrição da rede e procede com uma simulação. Ao final, um sinal de ack é enviado ao sistema (através da rota de sinais coms) informando que uma simulação chegou ao final. O resultado da simulação é também armazenado na base de associação.

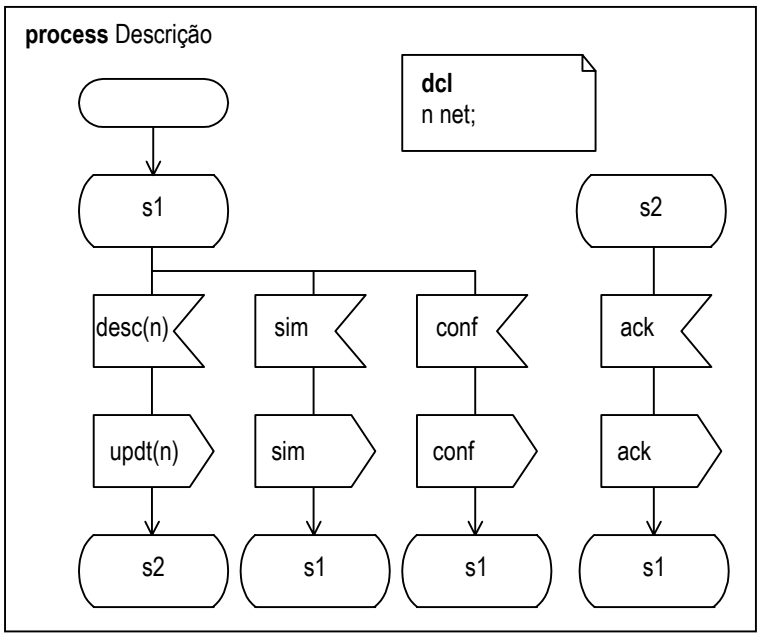

FIGURA 4.11 – Processo de descrição de uma rede

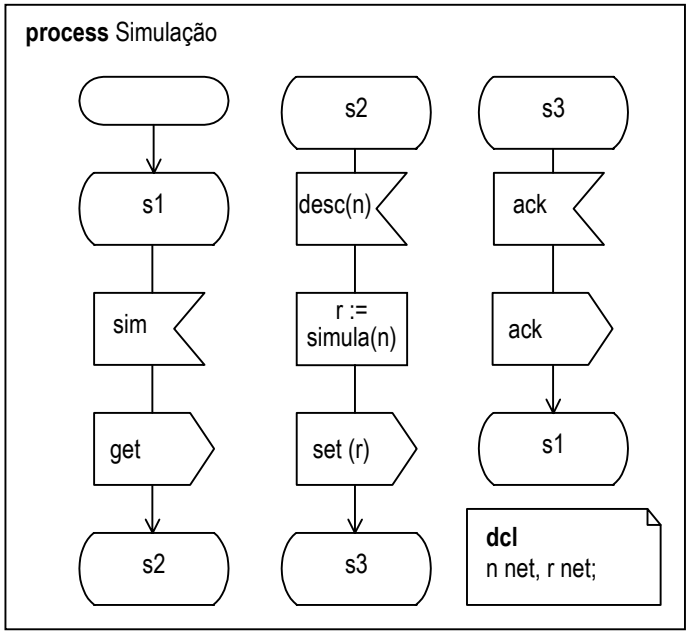

FIGURA 4.12 – Processo de simulação de rede

Por fim, um sinal para configuração de rede (conf) pode ser enviado ao bloco que faz com que o processo de configuração seja ativado (FIGURA 4.13). Este se comunica com a rede para programas os dispositivos através de sinais set e get.

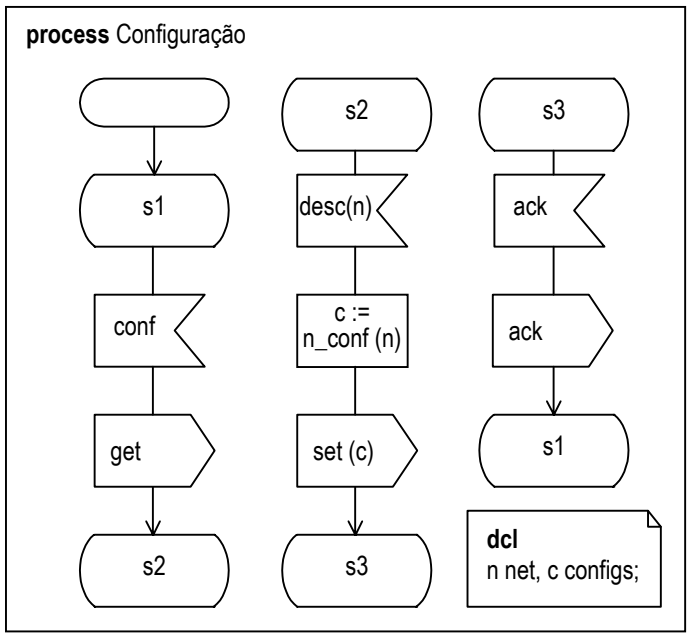

FIGURA 4.13 – Processo de configuração de rede

Análise

O bloco para análise de QoS (FIGURA 4.14) é composto por dois processos: programação e análise. É importante notar que o processo de programação é o responsável por criar novas instâncias de processos de análise. Assim, a análise de QoS acaba sendo executada por vários processos de análise.

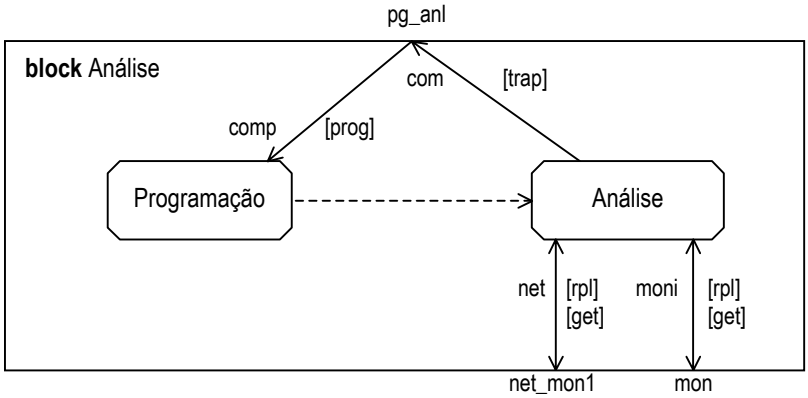

FIGURA 4.14 – Bloco de análise de QoS

O processo de programação recebe uma solicitação via sinal prog para que uma nova análise seja realizada. Nestas circunstâncias, o processo de programação cria uma nova instância do processo de análise e repassa as informações recebidas de programação a instância recém criada (FIGURA 4.15).

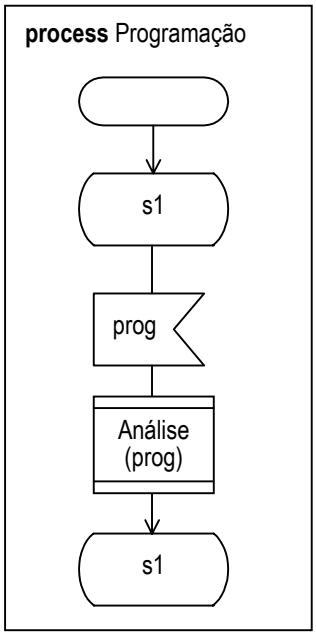

FIGURA 4.15 – Processo de configuração

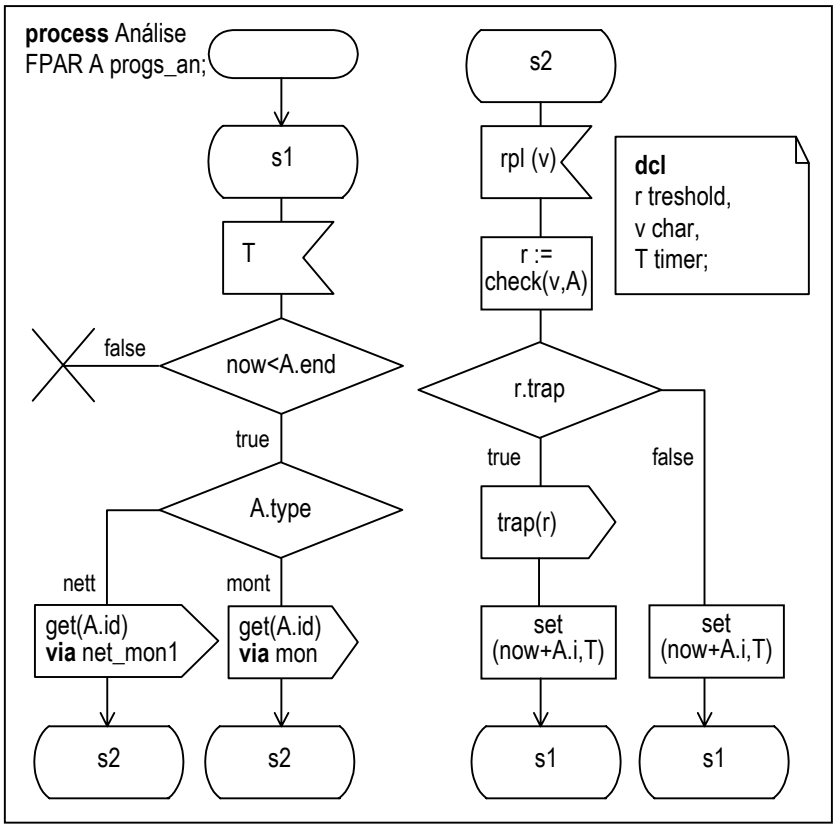

FIGURA 4.16 – Processo de análise de QoS

O processo de análise criado permanece ativo até o momento em que a descrição da análise a ser realizado informa seu final (A.end). Enquanto isso não ocorre, primeiramente o processo determina quem é o fornecedor dos dados para análise: a rede diretamente ou os processos de monitoração. Uma mensagem de solicitação (get) é

envia para a recuperação do dado a ser analisado. Se o dado fizer parte da rede, a rota de sinais net_mon1 é utilizada; se o dado fizer parte dos processos de monitoração a rota utilizada será a mon. A resposta (r) é então enviado de volta ao processo de análise que verifica se não existe nenhum problema (check). Caso um problema seja detectado o mesmo é notificado (trap) ao ambiente de gerência (FIGURA 4.16).

Identificação

O bloco para identificação de QoS, assim como o bloco de análise apresentado anteriormente, é composto de dois processos (FIGURA 4.17).

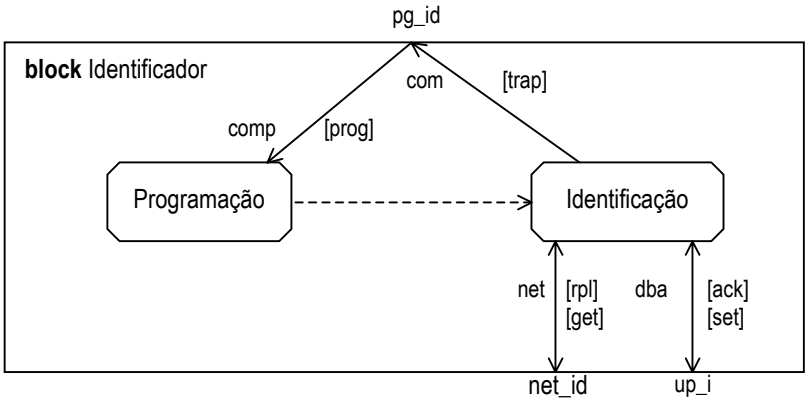

FIGURA 4.17 – Bloco de identificação de QoS

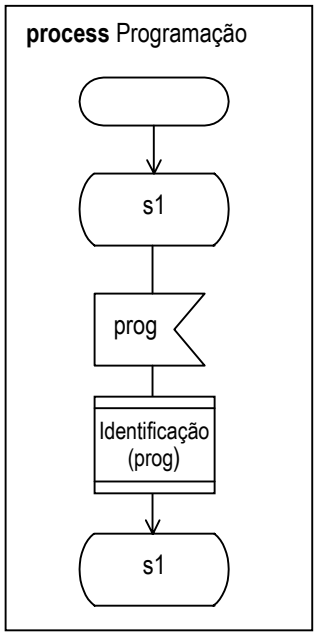

FIGURA 4.18 – Processo de programação de identificação de QoS

O processo de programação recebe uma solicitação via sinal prog para que uma nova identificação de QoS seja realizada. Nestas circunstâncias, o processo de programação cria uma nova instância do processo de identificação e repassa as informações recebidas de programação à instância recém criada (FIGURA 4.18).

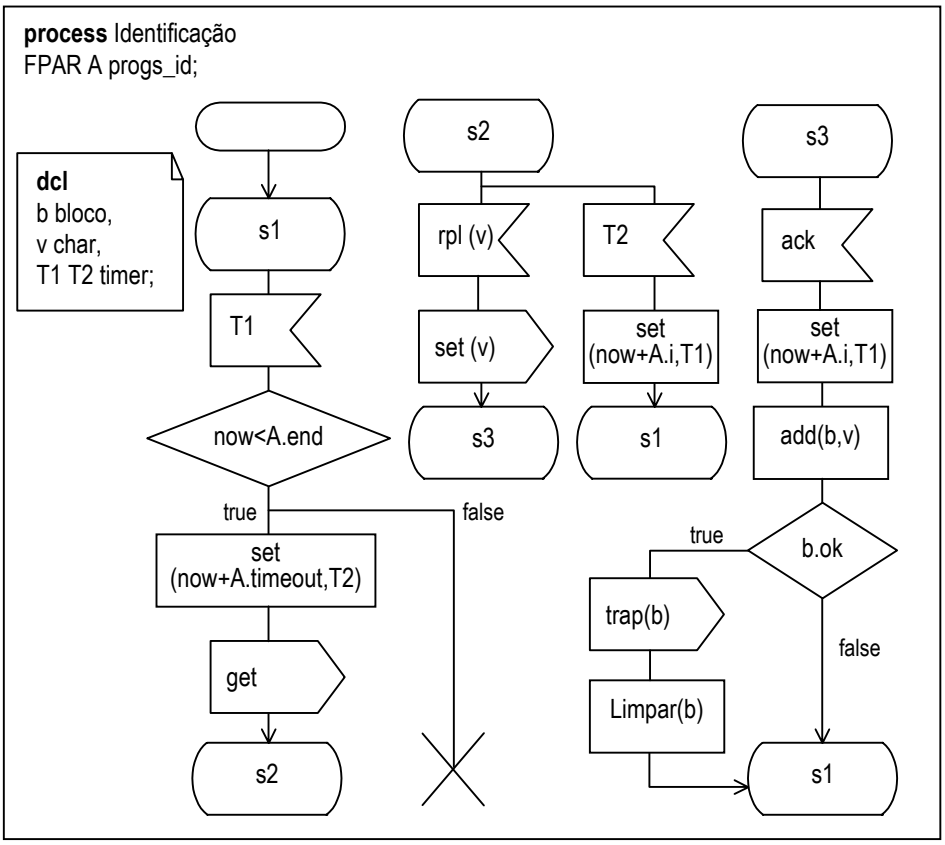

FIGURA 4.19 – Processo de identificação de QoS

O processo de identificação criado (FIGURA 4.19) permanece ativo até o momento em que a descrição da análise a ser realizada informa seu final (A.end). Enquanto isso não ocorre, o processo faz uma solicitação à rede de computadores a procura de novas características em relação ao fornecimento de QoS utilizando o sinal get. Um tempo máximo de espera de resposta é determinado (now+A.timeout) de forma que se a rede não responder o processo não entre em espera indefinida por uma resposta. Caso haja uma resposta um sinal set é enviado a base de associações para que a mesma cadastre o que foi recentemente retornado pela rede. Caso não haja nenhuma resposta o processo passa para a tentativa de uma nova identificação.

O próximo passo corresponde a determinação de blocos de notificações a serem enviados ao ambiente do usuário. Cada nova descoberta é incluída em um bloco lógico, e quando este atingir seu tamanho limite (b.ok) uma notificação acaba sendo enviada, via sinal trap ao ambiente de gerência. O bloco é então esvaziado para que novos elementos descobertos ocupem lugares em um bloco vazio.

Monitoração

Assim como nos blocos de análise e identificação, a monitoração de QoS conta também com dois processos distintos (FIGURA 4.20).

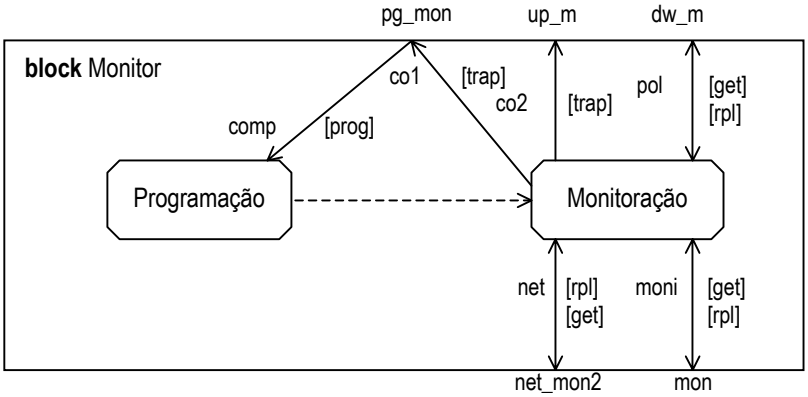

FIGURA 4.20 – Bloco de monitoração de QoS

Como antes, o processo de programação recebe solicitações de monitoração através de sinais prog (via rota de sinais comp) que desencadeiam a criação de novos processos de monitoração. Para cada solicitação um novo processo de monitoração exclusivo é utilizado. O bloco Monitor comunica-se com a rede monitorada (via rota net), com as bases de *status* e políticas (via rotas up_m e dw_m, respectivamente), além de receber também solicitações do bloco de monitoração através da rota moni. Neste último caso, a comunicação é fornecida para que os processos de análise sejam capazes de se comunicar com os processos de monitoração de forma que a análise de QoS possa ser realizada utilizando não apenas as informações colhidas na rede de computadores, mas também junto aos monitores de QoS ativos.

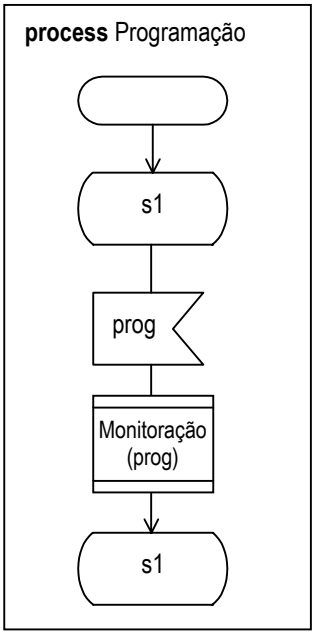

FIGURA 4.21 – Processo de programação da monitoração de QoS

O processo de monitoração inicialmente recupera a política a ser monitorada e permanece à espera do momento adequado para a realização das solicitações de monitoração à rede. Quando uma solicitação deve ser executada, a rede é contactada e a resposta obtida comparada com o as definições da política. Se uma degradação é observada, a base de *status* e o sistema de gerência são alertados através de mensagens trap enviadas pelas rotas de sinais co1 e co2, respectivamente.

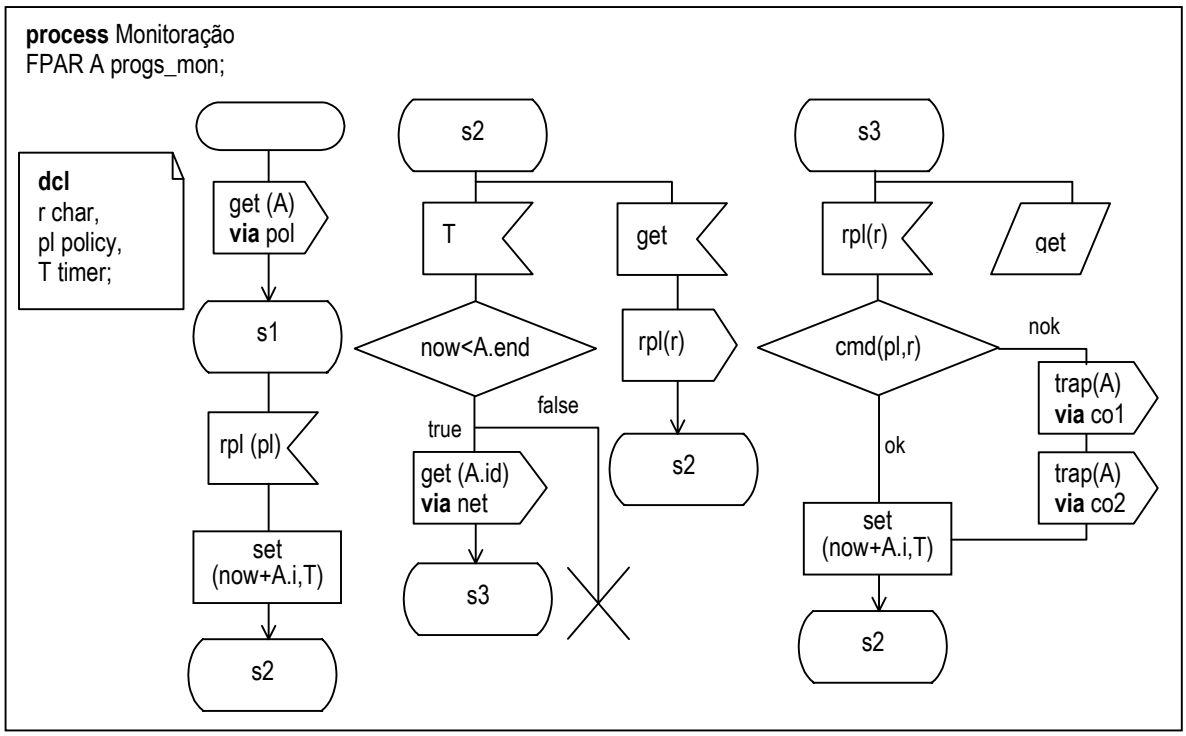

FIGURA 4.22 – Processo de monitoração de QoS

Enquanto espera pelo momento de novas solicitações à rede, o processo de monitoração pode receber sinais provindos dos processos de monitoração que solicitam dados sobre a última consulta realizada. Neste caso, os processos de monitoração respondem aos processos de análise e volta a esperar o momento correto para novas solicitações à rede. Caso um processo de análise tente contactar um processo de monitoração enquanto este estiver processando uma análise, o sinal de solicitação do processo de análise é armazenado para posterior processamento.

Consumidores de políticas

O bloco que define os consumidores de políticas também possui dois processos (FIGURA 4.23), sendo que novamente o processo de programação é o responsável por receber solicitações para que políticas sejam consumidas. Neste caso, o processo de programação (FIGURA 4.24) cria novas instâncias do processo *consome* que então é o responsável por implantar uma política em alvos da rede e verificar se a implantação ocorreu de forma adequada.

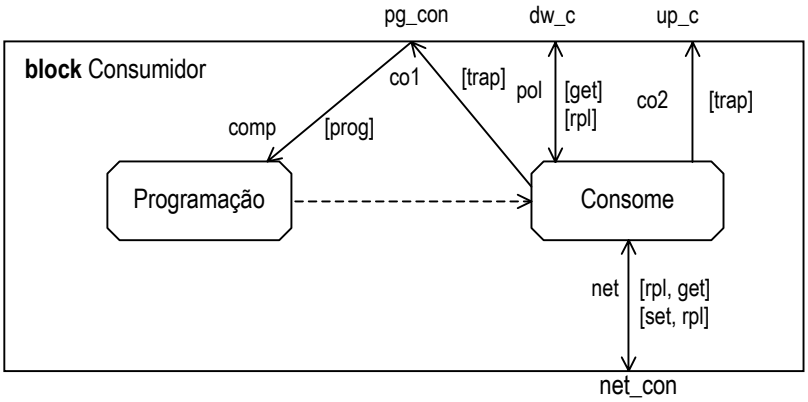

FIGURA 4.23 – Bloco dos consumidores de políticas

O bloco *consumidor* comunicação também com as bases de políticas e *status* através dos canais pol e co2, respectivamente.

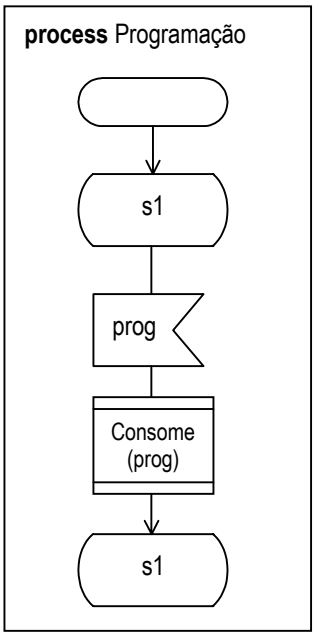

FIGURA 4.24 – Processo de programação de consumidores de políticas

Assim como nos processos de monitoração, a primeira ação do processo *consume* (FIGURA 4.25) é a recuperação da política de interesse na base de políticas. Para tal, um sinal get é enviado como solicitação da política a ser aplicada. A política então é ativada apenas no momento determinado em sua descrição. Se a política tiver seu tempo de vida máximo alcançado, então o processo de implantação é destruído, já que a política não será mais utilizada. Caso contrário, a política passa a ser implantada na rede, e o momento de desativação da política é observado. Quando uma política deixar de ser válida, mas futuramente ainda será utilizada, então a rede é desprogramada, mas um novo momento de ativação da política é definido. Se política não puder ser implantada, um sinal trap é gerado para indicar esta situação.

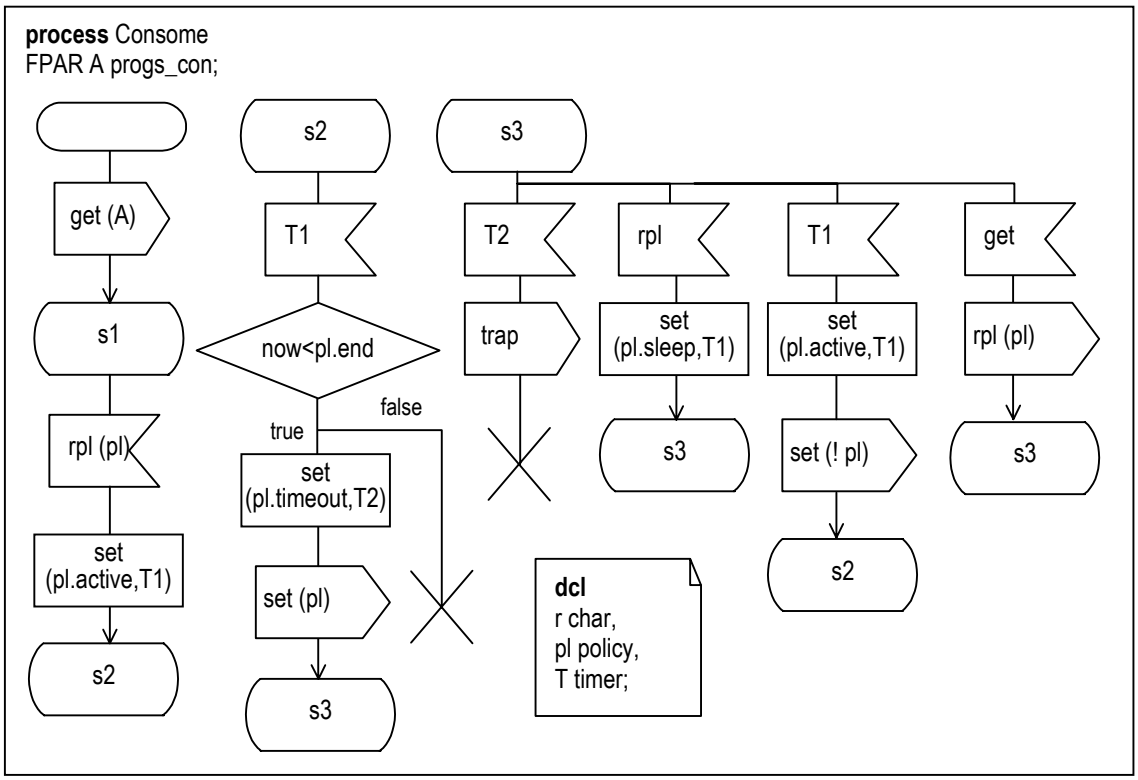

FIGURA 4.25 – Processo de implantação de políticas

Ambiente de gerência

O ambiente de gerência (FIGURA 4.26) é o bloco do sistema que possui o maior número de processos. Isso acontece porque o ambiente é o responsável por coordenar a execução dos outros blocos do sistema.

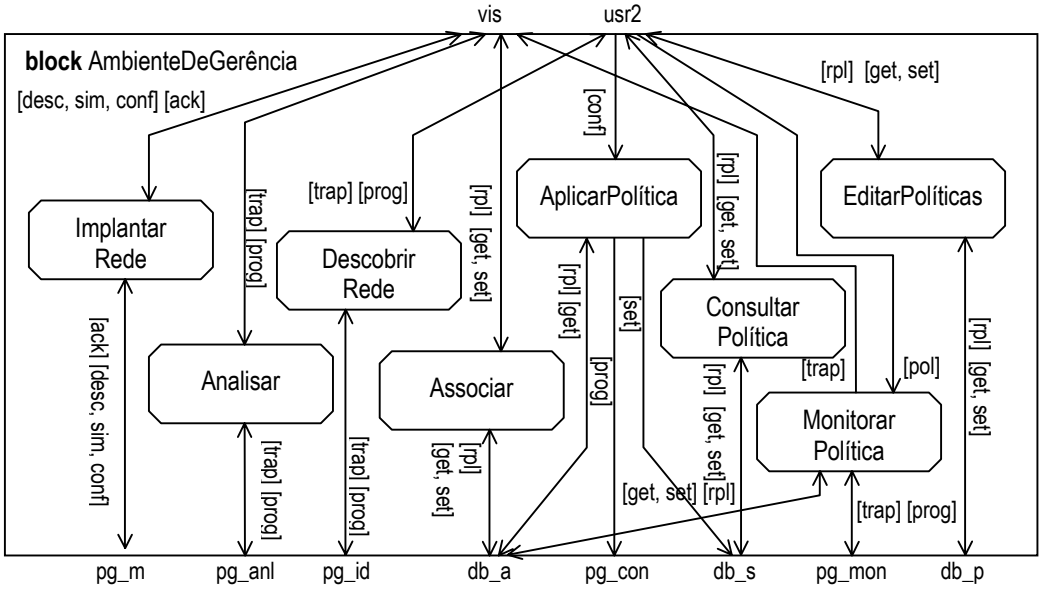

FIGURA 4.26 – Bloco do ambiente de gerência

Internamente, entretanto, os processos do bloco do ambiente de gerência são simples porque o maior parte do processamento é realizado por processos de outros blocos, apresentados anteriormente. Muitos processos do ambiente de gerência funcionam apenas como conectores entre o usuário do sistema (o administrador da rede) e os outros elementos da arquitetura. Por exemplo, os processos "ImplantarRede", "Analisar" (FIGURA 4.27), "DescobrirRede" e "Associar" (FIGURA 4.28) apenas repassam mensagens aos outros blocos do sistema.

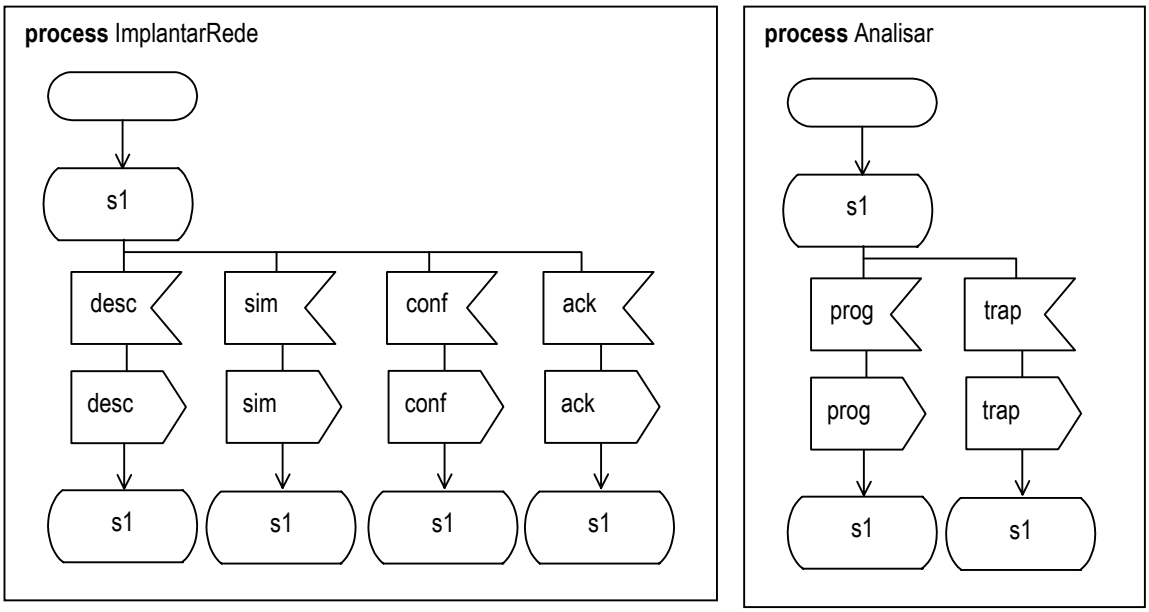

FIGURA 4.27 – Processos "ImpantarRede" e "Analisar"

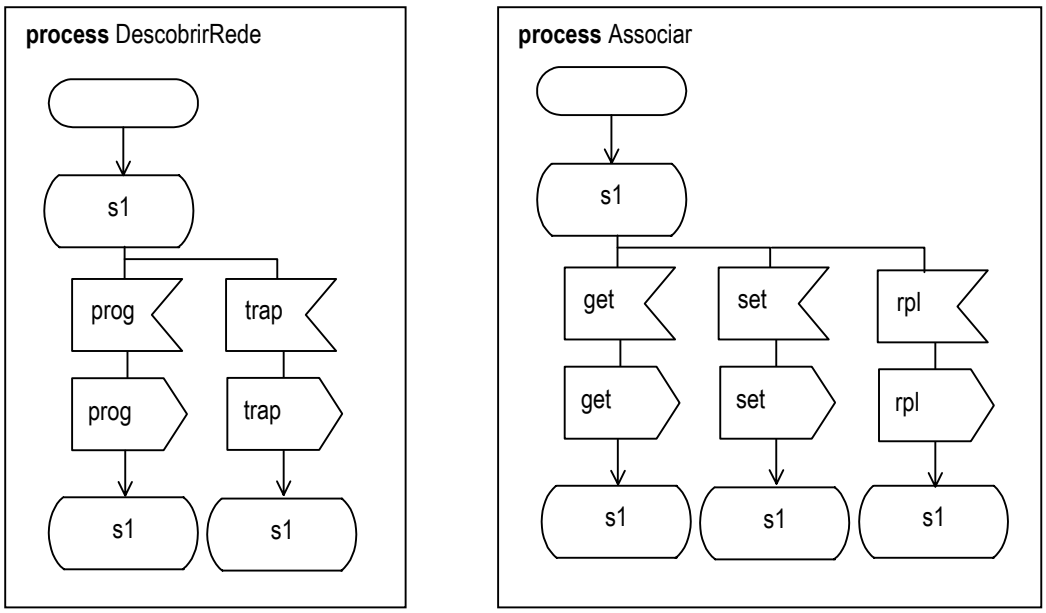

FIGURA 4.28 – Processos "DescobrirRede" e "Associar"

O processo “AplicarPolíticas” (FIGURA 4.29) é o mais complexo, pois envolve a determinação de consumidores de políticas a serem utilizados, a partir de um conjunto de alvos. A seguir, uma nova entrada na base de *status* deve ser determinada, e os consumidores encontrados devem ser programados.

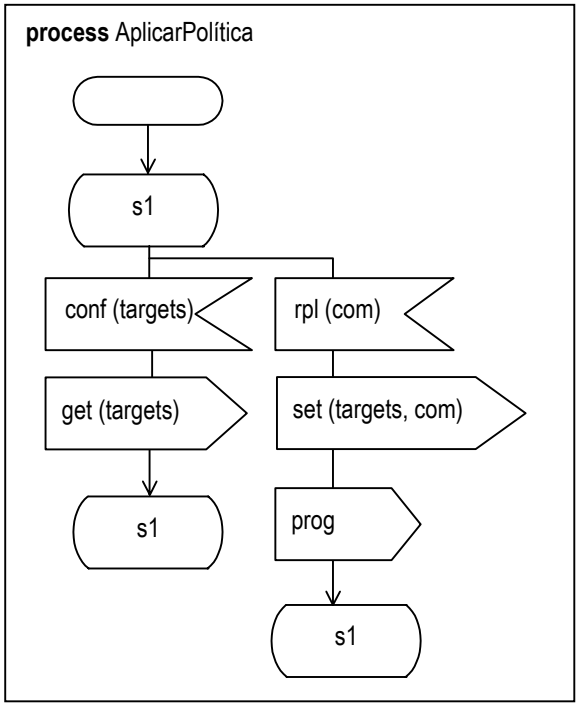

FIGURA 4.29 – Processo “AplicarPolíticas”

Os processos “ConsultarPolítica” e “EditarPolíticas” também são simples e apenas repassam sinais. Apesar da simplicidade, a inclusão de outros processamentos dentro de cada uma destes processos poderia implementar níveis de abstração maiores, como discutido na seção 4.11.

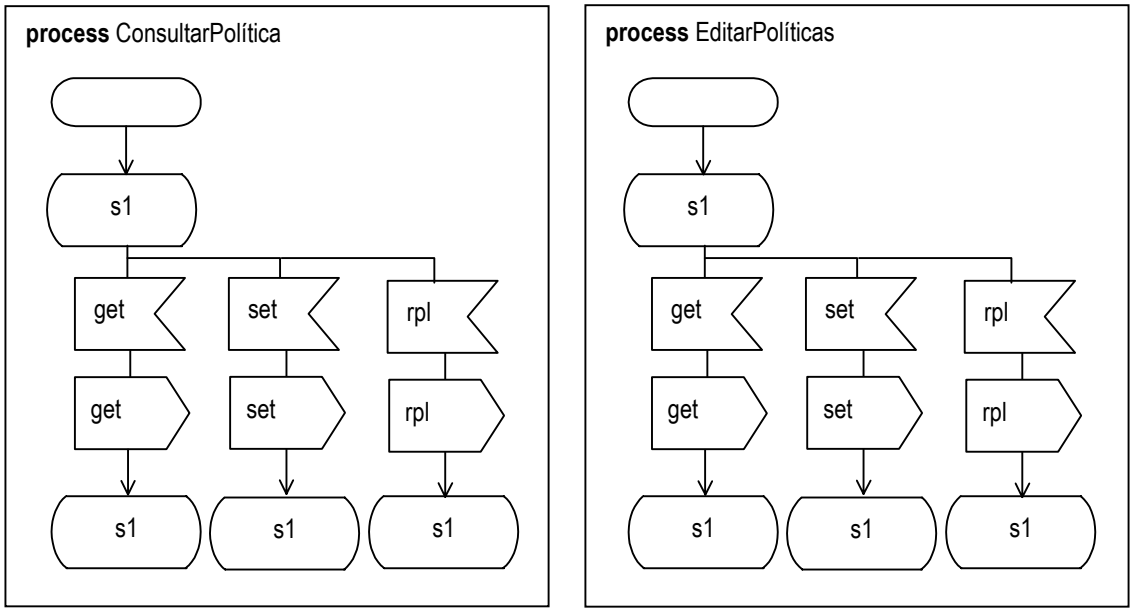

FIGURA 4.30 – Processos “ConultarPolítica” e “EditarPolíticas”

Finalmente, o processo “MonitorarPolítica” recupera os monitores de QoS associados a uma política e repassa a estes a programação para que os mesmos passem a verificar o QoS real da rede. Quando degradações são observadas, estas são repassadas à visualização de QoS.

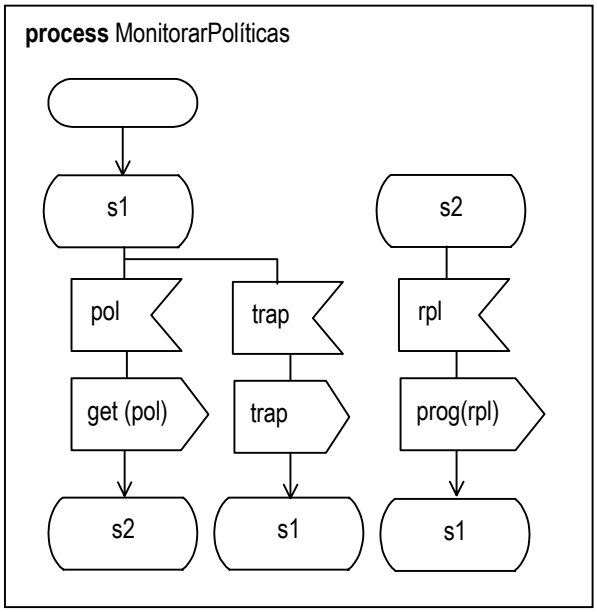

FIGURA 4.31 – Processo “MonitorarPolíticas”

5 Análise do modelo proposto

A necessidade de integração das tarefas de gerência de QoS motivou a criação do modelo para gerência integrada apresentado no capítulo anterior. Os elementos do modelo foram descritos e dois exemplos serviram como bases para discussões sobre a aplicação do modelo em redes que necessitavam de gerenciamento de QoS.

Neste capítulo serão analisados outros aspectos do modelo proposto. O objetivo deste capítulo é fornecer um estudo mais aprofundado do modelo de gerência utilizando também resultados obtidos a partir de um protótipo.

5.1 O modelo proposto e as tarefas de gerência de QoS definidas

Uma questão crítica, e que norteia toda a definição do modelo é a habilidade de fornecer suporte às seis tarefas de gerência de QoS definidas no capítulo 3. Como as tarefas definidas na prática já podem ser realizadas atualmente com soluções existentes, o modelo deve fornecer um suporte integrado às tarefas, de forma a facilitar a execução das mesmas por parte do administrador da rede.

Os elementos intermediários definidos (consumidores de políticas, monitores de QoS e identificadores de alvos), além de permitirem uma abstração das complexidades dos alvos existentes na rede gerenciada, apresentam suporte à manutenção, monitoração e descoberta de QoS. Estas três tarefas são executadas pelos elementos intermediários porque são as tarefas que exigem uma interação maior com os dispositivos e alvos da rede. Logo, é menos apropriado pensar em se ter um consumidor de políticas, ou identificador de alvos junto ao ambiente de gerência, embora seja possível.

Os elementos que tipicamente encontram-se junto ao ambiente de gerência são então justamente aqueles onde a interação direta com a rede é menos acentuada, ou que a interação com o administrador mereça mais atenção. Assim, o suporte à implantação e análise de QoS (que exigem menos interações com a rede), e à visualização de QoS (que exige maior interação com o administrador) é encontrado junto ao ambiente de gerência do modelo.

Com estes seis elementos, as seis tarefas de gerência definidas encontram suporte adequado. A integração das tarefas, por sua vez, é alcançada através do compartilhamento, entre os elementos do modelo, de recursos de gerência comuns. O principal recurso compartilhado, e principal suporte à integração, são de bases de dados comuns. Em soluções padrão, cada *software* utilizado acaba implementando a sua própria base de dados. No modelo apresentado, entretanto, as bases utilizadas são acessadas por todos os elementos.

Além das bases, os dados propriamente ditos são comuns aos elementos. As políticas são o exemplo mais significativo neste caso. Definidas pelo administrador da rede, e armazenadas na base de políticas, as políticas são utilizadas pelos consumidores para programar a rede de acordo com os objetivos definidos, e pelos monitores de QoS para verificação da correta operação dos serviços em relação ao QoS. As políticas podem também ser utilizadas pelo processo de análise de QoS que pode indicar as

políticas que freqüentemente apresentam maior degradação, por exemplo. Outros dados compartilhados são também: as informações relativas aos alvos e dispositivos cadastradas na base de associações; o estado de uma política aplicada em um alvo; entre outros.

A integração também é resultado da coordenação, por parte do ambiente de gerência, das tarefas executadas pelos elementos do modelo. Por exemplo, a automatização da determinação de quais são os monitores de QoS necessários para a verificação de uma política só é possível porque o ambiente de gerência executa consultas adequadas à base de associações (a partir de uma política o ambiente de gerência determina quais os alvos associados, e com uma consulta a cada alvo são determinados os monitores de QoS adequados).

Como visto na seção 4.11 (segundo exemplo), quanto maior o nível de abstração encontrado no ambiente de gerência, menos complexas são as tarefas executadas pelo administrador. O fornecimento de abstrações também é um fator que promove a integração das tarefas, já que o administrador da rede passa a executar operações admitindo que o sistema de gerência irá coordenar todos os elementos envolvidos de forma a atingir as metas estabelecidas pelo administrador. A FIGURA 4.8 (determinação de monitoração através de uma linguagem de alto nível) é um exemplo de integração de tarefas de gerência de QoS (manutenção e monitoração) através do uso de abstrações implementadas no ambiente de gerência.

Pode-se dizer então que o suporte às tarefas de gerência de QoS é conseguido com a existência de elementos específicos que suportam cada uma dessas tarefas, enquanto que o suporte integrado a tais tarefas é o resultado de:

- Bases de dados comuns (políticas, *status* e associações);
- Esquemas de dados comuns (as políticas, as informações dos alvos, etc.);
- Coordenação de tarefas no ambiente de gerência (determinação de monitores associados às políticas, descoberta de QoS "alimentando" a implantação de QoS, etc.);
- Abstrações de linguagem implementadas no ambiente de gerência.

5.2 Aplicação do modelo em redes diferentes

Um dos requisitos na definição de modelo de gerência era a necessidade de o modelo ser genérico de forma que pudesse ser utilizado no gerenciamento de redes tão diferentes como são, por exemplo, as redes com tecnologia ATM e redes IP. Em relação a este aspecto, nesta seção serão apresentadas duas possíveis utilizações do modelo em redes diferentes.

Na primeira análise é apresentada uma arquitetura para a gerência de redes ATM. A segunda análise é apresentada através de uma arquitetura utilizada para a gerência de redes IP com algum tipo de suporte a QoS (serviços integrados ou diferenciados, por exemplo).

Arquitetura para gerência de QoS em redes ATM

Em relação ao fornecimento de QoS, provavelmente as redes ATM são aquelas que implementam serviços mais adequados. Isso acontece principalmente porque serviços com QoS foram introduzidos em redes ATM desde sua concepção. A solução para fornecimento de QoS em redes ATM é formada por um conjunto de normas bem definidas cujo conjunto de serviços oferecidos permitem o tráfego de dados e informações multimídia e de tempo-real de forma adequada. Comparativamente com as redes IP (e respectivas soluções de fornecimento de QoS), uma rede ATM é mais "confiável" porque os serviços oferecidos são padronizados e únicos.

A aplicação do modelo de gerência proposto em uma rede ATM permitiria a um administrador inicialmente planejar sua rede via implantação de QoS. Como os serviços ATM são únicos, fica fácil a escolha da configuração da rede. A implantação, neste caso, pode auxiliar mais efetivamente na definição da topologia da rede a ser implantada através de simulações.

A maior parte dos equipamentos ATM suporta o protocolo SNMP, o que em muitos casos permite não apenas a monitoração dos equipamentos, mas também a sua configuração. Entretanto, freqüentemente protocolos proprietários também podem ser utilizados. Nestas situações, políticas são utilizadas para abstrair as diferenças entre os equipamentos. Os consumidores de políticas são então utilizados para a programação dos dispositivos ATM.

Uma questão crítica em redes ATM é o controle de admissão. Os *switches* ATM executam o controle de admissão utilizando protocolos de sinalização para decidir se novos fluxos (em redes ATM um fluxo faz parte de um circuito virtual) podem ser suportados. O controle de admissão pode se favorecer das políticas para decidir se um determinado circuito virtual pode ser estabelecido de acordo com os objetivos da rede. Neste caso, assim que um *swtich* recebesse uma solicitação de estabelecimento de circuito virtual, além das verificações padrão executadas pelo *switch*, uma comunicação com um consumidor de políticas seria necessária para determinar, de acordo com os objetivos da rede, se o circuito virtual realmente poderia ser estabelecido. Este mecanismo de comunicação originalmente não existe em uma rede ATM, mas poderia ser implementado através da utilização do protocolo COPS.

Por serem baseadas principalmente em *switches*, redes ATM têm um grande potencial de análise de tráfego através da utilização de RMON e RMON2. Muitos equipamentos ATM atualmente já implementam as duas MIBs, e isso permite que se faça uma monitoração de *swtiches* remotamente. Outras MIBs ATM específicas acabam por completar as definições RMON e RMON2 para que outras informações sobre a utilização dos recursos de rede possam ser fornecidos [GRA 2000f]. Com tais informações, monitores de QoS podem ser utilizados para se fazer a verificação do comportamento da rede. Entretanto, como o fornecimento de QoS em redes ATM é muito mais "seguro" que em redes IP, os processos de monitoração de QoS poderiam até mesmo serem dispensados. Nesta abordagem, confia-se que a rede será realmente capaz de fornecer serviços com as garantias necessárias, e um processo de monitoração seria dispensável.

Ainda que os processos de monitoração possam não ser usados em uma rede ATM, o uso de processos para análise de QoS continua sendo necessário. Os processos de análise trabalham em escalas temporais maiores e possibilitam, por exemplo, a

percepção de que um determinado segmento de rede deve ser atualizado com maior banda porque o número de solicitações de circuitos virtuais negadas está crescendo de uma forma inesperada. Neste contexto, as MIBs RMON, RMON2 e MIBs de monitoração proprietárias são úteis porque permitem aos processos de análise de QoS verificarem o histórico da utilização da rede e detectarem então a necessidade de novos recursos, como citado.

Assim como a monitoração de QoS, a descoberta de QoS é uma tarefa que pode ser menos importante numa implementação do modelo proposto em uma rede ATM. O principal motivo para isso é que os serviços disponibilizados em uma rede ATM são capazes de coordenar-se entre si e perceber a existência, por exemplo, da rotas alternativas mais adequadas em uma rede. Logo, neste sentido, a descoberta de QoS também pode ser dispensada de uma implementação em redes ATM.

Por fim, a visualização de QoS em redes ATM pode apresentar informações sobre os circuitos virtuais estabelecidos, suas características e rotas. Pode mostrar o resultado da análise de QoS e apontar dispositivos críticos (por exemplo, que estão negando muitos estabelecimentos de circuitos virtuais). A FIGURA 5.1 apresenta uma arquitetura possível para a utilização do modelo de gerência de QoS em uma rede ATM.

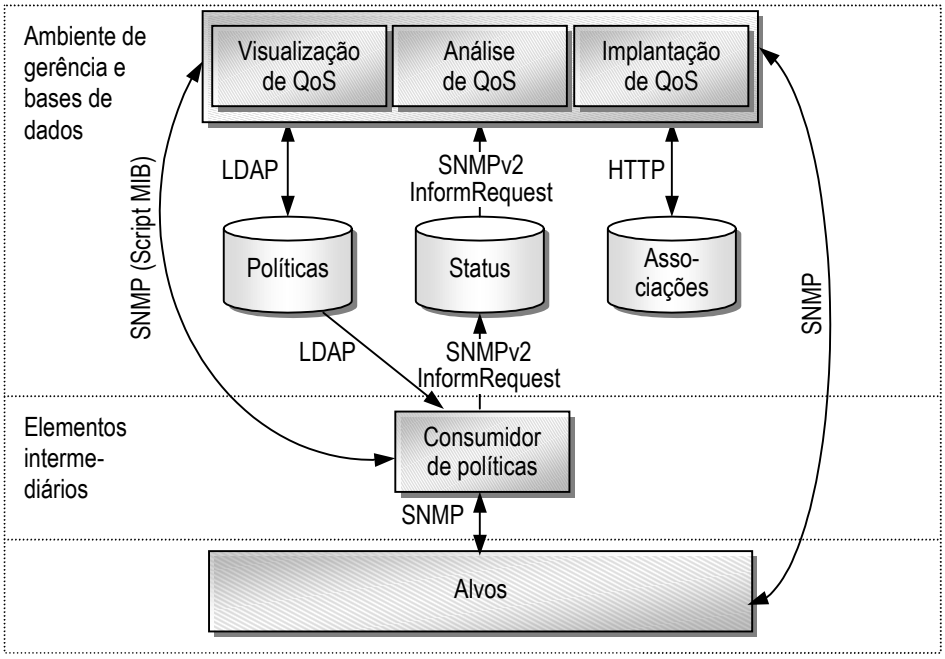

FIGURA 5.1 – Arquitetura para gerência de redes ATM

A comunicação entre todos os elementos da arquitetura é realizada através de circuitos virtuais permanentes específicos para o tráfego de gerenciamento. A base de políticas é implementada em um servidor de diretórios LDAP que o consumidor de políticas acessa para recuperar as políticas que regem o comportamento da rede. O acesso ao consumidor de políticas pelo ambiente de gerência é feito via SNMP através da transferência de *scripts*. O acesso aos alvos pelos consumidores de políticas pode ser feito via SNMP ou por protocolos proprietários, quando o SNMP se mostrar insuficiente. O SNMP também é utilizado como meio de acesso aos alvos para que o ambiente de gerência, principalmente através dos processos de análise de QoS, possa verificar o comportamento a longo tempo da rede (já que o comportamento em curto

espaço de tempo não é analisado pois não existiriam monitores de QoS: a arquitetura ATM é tida neste caso como confiável).

A base de associações é implementada utilizando-se um banco de dados padrão, mas sua interface externa é disponibilizada via HTTP. Um mecanismo PHP4, por exemplo, pode ser utilizado para permitir a inclusão, exclusão e alteração de dados da base. Por fim, a base de *status* também é implementada através de um banco de dados padrão, mas acessada via mensagens *InformRequest* do SNMPv2.

QAME (QoS-Aware Management Environment) para redes IP

O fornecimento de QoS em redes IP é mais complexo que o fornecimento de QoS em redes ATM pelo simples fato de que as redes IP não foram pensadas para fornecerem QoS. O objetivo principal original de uma rede IP era a robustez da rede em relação à entrega de mensagens. Por conta disso, rotas alternativas, uso de datagramas, e a falta de serviços com conexão no nível de rede são as principais características de uma rede IP padrão.

Com o advento da Internet e a evolução das tecnologias Web, o IP acabou sendo utilizado para a transmissão de conteúdos diversos, muitas vezes inadequados ao protocolo. O enorme sucesso da Internet é a prova final da escalabilidade do protocolo. Entretanto, com a necessidade crescente de QoS para a transmissão de dados com restrições temporais (VoIP [MIN 98], videoconferência, telemedicina, etc.) os serviços IP passaram a ser revisados, e propostas para o fornecimento de QoS diversas começaram a surgir.

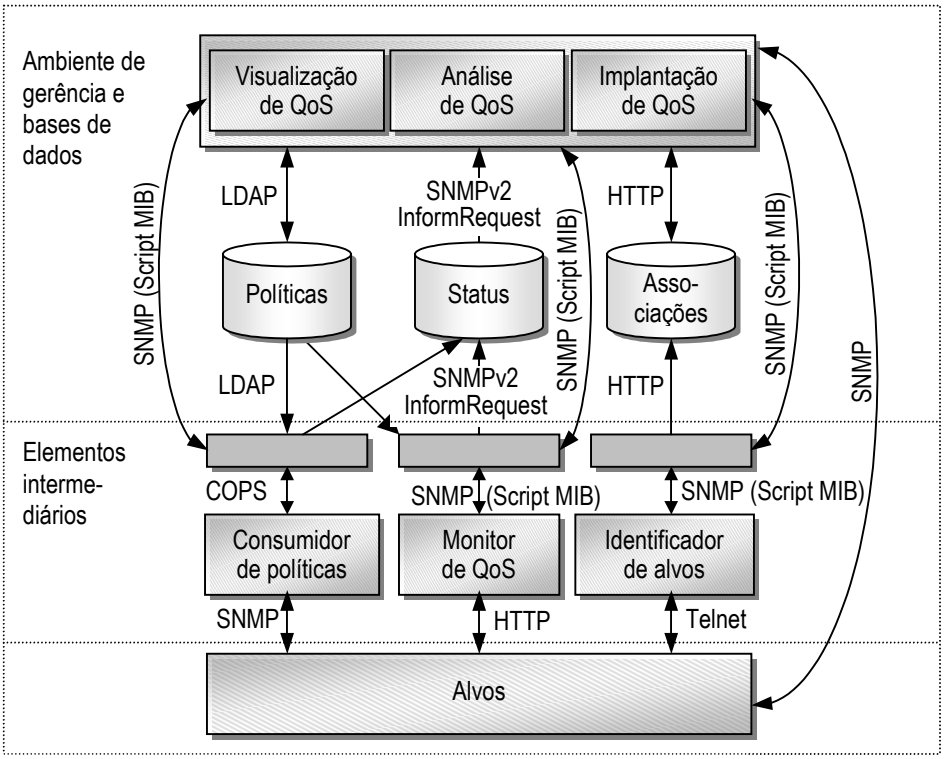

FIGURA 5.2 – Arquitetura QAME para gerência de QoS em redes IP

O resultado final é que atualmente uma rede IP é tipicamente heterogênea não apenas em relação aos seus equipamentos, mas também em relação aos seus serviços e o

suporte ao fornecimento de QoS. Como conseqüência direta, uma rede IP é mais complexa de ser gerenciada, em relação ao QoS, que uma rede ATM. Neste sentido, uma rede IP, sendo gerenciada através do modelo proposto pode apresentar um sistema de gerência que possui a arquitetura da FIGURA 5.2

A arquitetura apresentada é a implementada no ambiente QAME (*QoS-Aware Management Environment*) [GRA 2001b] [GRA 2001e] e de forma geral segue as definições do modelo de gerência proposto. Na FIGURA 5.2 são apresentados também os protocolos utilizados na implementação do protótipo, e as interações entre os elementos. A interação entre os elementos do ambiente QAME é aquela definida pelo modelo. A comunicação dos consumidores de políticas e monitores de QoS com a base de políticas é realizada através do modelo *pull*.

O sistema está baseado em MySQL [YAR 99] para a implementação das bases de *status* e associações, e em OpenLDAP [UNI 96] para a implementação da base de políticas. Um consumidor de políticas para serviços diferenciados [COE 2001] e um monitor de QoS [RIB 2001a] correspondente foram construídos. Os processos de descoberta de QoS foram implementados em Java. Os elementos foram dispostos em uma rede e um ponto importante em relação à localização dos elementos é a interface entre o ambiente de gerência e os elementos intermediários (identificadores de alvos, monitores de QoS e consumidores de políticas). Esta interface é apresentada na FIGURA 5.2 através de retângulos cinza que conectam o ambiente do usuário com os elementos intermediários. Esta interface pode estar localizada junto aos elementos intermediários, se tais elementos implementarem a interface. Caso contrário, as interfaces estarão localizadas no ambiente do usuário e procedem com a tradução de requisições para comandos específicos dos dispositivos. Esta separação entre a implementação dos elementos e interfaces é importante porque dispositivos IP mais modernos podem implementar elementos intermediários com interfaces de comunicação diferentes daquelas usadas pelo QAME. Neste caso uma tradução de interfaces é necessária para permitir o uso de elementos localizados internamente aos dispositivos.

O conjunto de protocolos utilizados é amplamente disponível, muitas vezes através de soluções abertas. Os elementos internos ao ambiente do usuário foram implementados utilizando Flash [MAC 2001a] (visualização de QoS) e MRTG [OET 2001] (análise de QoS). A implantação de QoS é o item mais crítico. Neste caso, o uso do *software* NS é adequado porque fornece um ambiente de simulação importante [NS 2001]. Mais detalhes sobre o ambiente são apresentadas na seção 5.6.

5.3 Modularidade do modelo

Na seção anterior, através das arquiteturas de gerência para ATM e IP, foi possível verificar quão genérico o modelo pode ser. Essa característica é importante para que o modelo possa se acomodar em redes diferentes. A diversidade de aplicações e serviços na rede, entretanto, permite que um conjunto de variáveis a serem suportadas na gerência seja ainda maior que o existente atualmente. Nestas condições, o modelo apresentado deve se adaptar às necessidades dos administradores de forma adequada.

A adaptabilidade do modelo em situações diferentes das normais pode ser realizada de diversas formas. A criação de novas políticas é a forma mais básica. Porém, em situações mais específicas, o uso de novos módulos pode passar a ser necessário

como forma de complementação do ambiente de gerência existente. A utilização de novos módulos deve, obviamente, ser feita também de forma integrada à solução, para que o administrador da rede tenha um ambiente único de acesso às várias funcionalidades existentes, inclusive as novas funcionalidades incorporadas.

O modelo proposto permite a inclusão de novas funcionalidades através da modularidade existente. Novos módulos criados podem ser incorporados, e obter comunicação com os outros módulos existentes através de dois modelos principais: indiretamente, via bases de dados, e diretamente, via API de programação e/ou protocolos de comunicação.

Integração indireta via bases de dados

A integração indireta via bases de dados é possível quando os novos módulos criados recebem acesso às bases e podem proceder com leituras e gravações. Como todos os outros elementos do modelo tem a integração também baseada no uso comum das bases, um novo módulo inserido é capaz de se comunicar com os outros módulos já existentes utilizando as bases.

Teoricamente, a base que deve sofrer menos alterações por novos módulos é a base de políticas, pois as funcionalidades criadas originalmente devem ser suficientes em uma implementação. Entretanto, por exemplo, um novo módulo de criação de políticas poderia ser adicionado para dar suporte ao gerenciamento de segurança (não coberto nesta tese). Outro exemplo seria a criação de um módulo para gerência de contabilização, onde um usuário teria acesso à rede apenas em períodos específicos do dia. Nestas duas situações, módulos adicionais permitiriam ao administrador criar políticas relacionadas não apenas ao QoS. As políticas seriam armazenadas pelos módulos na base de políticas e aplicadas por novos consumidores de políticas capazes de traduzir as novas políticas.

Com novos módulos criados para suportar outras políticas, também é esperada a criação de outros módulos capazes de verificar a efetividade de tais políticas. Tais módulos (que no caso do QoS são os monitores de QoS) acabarão tendo acesso à base de *status* para atualizar o estado de um política em observação. Logo, a base de *status* também permitirá uma integração indireta.

Por fim, a base de associações fornece o ponto de maior integração indireta do modelo, já que possui os relacionamentos entre informações mais importantes do ambiente. Vários módulos poderiam ser criados neste contexto. Por exemplo, no modelo original o administrador da rede é responsável por associar os consumidores de políticas e monitores de QoS aos alvos encontrados. Um módulo novo, utilizando-se de técnicas de inteligência artificial, poderia criar associações iniciais desses elementos na base de associações que posteriormente seriam validadas, se necessário, pelo administrador. Novos módulos, utilizando tecnologias diferentes, poderiam ser utilizados para descobrir dispositivos que normalmente não seriam descobertos pelos processos padrão. Novamente, a base de associações seria acessada pelos novos módulos que informariam os dispositivos que anteriormente não poderiam ser descobertos.

Uma questão importante em relação à integração indireta é que possivelmente algumas informações necessárias aos novos módulos podem não existir nas bases originais. Felizmente, a inclusão de novos campos em tabelas não tende a causar

problemas para os módulos que já utilizavam as bases anteriormente. Por exemplo, um campo poderia determinar se um roteadores possui internamente filtros para funcionar como *firewall*. Como a gerência de segurança não era o foco do modelo, originalmente este campo não existiria, mas um processo de descoberta de *firewalls* e similares necessitaria desse campo. Nesse caso, a tabela que descreve os dispositivo teria um campo a mais incluído, e todos os dispositivos até então cadastrados receberiam por padrão o valor *false* para o campo, sem prejuízo para o funcionamento dos módulos já existentes.

Integração direta via API de programação e/ou protocolos de comunicação

A integração indireta usando acesso às bases de dados, como visto anteriormente, é possível. Entretanto, muitas vezes uma comunicação direta com um módulo já existente é necessária. Por exemplo, supondo que um novo módulo do sistema foi criado para a detecção de um ataque (novamente, gerência de segurança), e que a indicação de tal ataque deve ser feita pela alteração da cor utilizada no dispositivo atacado no mapa da rede, uma comunicação direta com os processos de visualização deve ser fornecida. Neste caso, deve existir uma API que permite ao novo módulo de detecção de ataque informar aos processos de visualização para mostrarem o computador atacado na cor vermelha, por exemplo.

Para que esse tipo de integração seja possível, duas soluções diferentes podem ser utilizadas: APIs de programação ou protocolos de comunicação. Antes de se entrar em detalhes sobre estas possibilidades, é importante notar que a integração direta depende fundamentalmente das facilidades que as implementações dos módulos "padrão" do modelo disponibilizam. Isso significa dizer que se um módulo original não permitir nenhum tipo de comunicação externa, ainda que tal módulo siga estritamente as definições do modelo, não existirá um mecanismo de integração disponível para este módulo.

A integração direta mais comumente encontrada é aquela que utiliza protocolos de comunicação para realizar a troca de informações entre os elementos existentes. Neste caso, cada elemento (ou módulo) do modelo fornece um meio de entrada ou ativação de suas funções ao mundo externo. Novos módulos implementados utilizam então estas facilidades dos outros módulos para estabelecerem comunicações. Na seção 5.2, a comunicação com os elementos intermediários da arquitetura QAME era realizada através da Script MIB. Novos módulos criados também poderiam utilizar a Script MIB para estabelecer conversações com os módulos "originais" do modelo. Por exemplo, um monitor de QoS diferente poderia comunicar-se diretamente com um consumidor de políticas para informar que uma das políticas monitoradas está apresentando degradações.

A interação entre módulos menos simples, em relação à utilização de protocolos de comunicação, é aquela que necessita ser realizada com os elementos internos ao ambiente de gerência do modelo (módulos para visualização, análise e implantação de QoS). De certa forma, o ambiente de gerência "esconde" tais elementos através dos pontos de acesso a serviços. Como tais pontos são restritos a algumas comunicações (notificações, transferência de políticas, etc.) o acesso direto aos elementos internos ao ambiente de gerência possivelmente acaba sendo necessário freqüentemente. Neste caso, ou a implementação explicitamente fornece um protocolo para comunicação com tais

elementos, como acontece com os elementos intermediários do modelo, ou APIs de programação devem ser utilizadas.

A idéia principal das APIs de programação é permitir que os módulos existentes possam ser expandidos em relação às suas funcionalidades. Um exemplo típico de expansão de módulo é o mecanismo de recebimento de *traps* SNMP muito freqüentemente encontrado em plataformas de gerência. Tipicamente, o mecanismo recebe *traps* e imprime seu conteúdo na saída padrão. Entretanto, uma expansão pode fazer com que as *traps* sejam direcionadas a outros processos de análise, enviadas por e-mail, ou qualquer outra opção, limitada apenas pela expansão do módulo. Entre as expansões possíveis, pode-se inclusive implementar protocolos de comunicação que acabam permitindo a interação externa nos mesmos modos descritos anteriormente.

Na verdade, existe ainda um terceiro modo de integração possível, mas menos factível: a reprogramação de módulos a partir de seus códigos fonte. Nesta solução, praticamente qualquer tipo de comunicação pode ser implementada, utilizando qualquer protocolo, já que se está de posse do código fonte de um módulo. Apesar de ser possível, já que módulos com código aberto estão se tornando cada vez mais disponíveis, a solução não é factível porque dificilmente um administrador de rede irá despender tempo com uma reimplementação.

5.4 Complexidade do modelo

A complexidade do modelo proposto é variável de acordo com o aspecto de interesse. Nesta seção será discutida a complexidade do ponto de vista do administrador de rede em relação à implantação e utilização do modelo. Será ainda verificada a complexidade do ponto de vista dos fornecedores de dispositivos de rede que fornecem suporte à gerência de rede internamente em seus equipamentos.

Complexidade de implantação

Comparativamente, o modelo proposto é mais complexo que o modelo de gerenciamento padrão centralizado amplamente adotado. E o motivo para isso é simples: para que uma rede passe a ser gerenciada através do modelo, um conjunto maior de atividades deve ser executado antes do início da aplicação do modelo.

Por questões de comparação, as arquiteturas de gerência centralizada (citada no parágrafo anterior), hierárquica e distribuída [LEI 96] serão abordadas. Diversas variações podem existir, mas por simplificação apenas estas três arquiteturas serão utilizadas. Uma breve caracterização passa a ser importante neste contexto. A arquitetura centralizada possui uma estação de gerência central que acessa os dispositivos de interesse diretamente. A rede gerenciada normalmente é mapeada através de um mecanismo centralizado de descoberta e passa a ser imediatamente gerenciada. Nenhum elemento intermediário é necessário, exceto *proxies* SNMP, mas que são encarados como dispositivos finais pelo sistema de gerência. Na arquitetura hierárquica existirão gerentes intermediários que executam tarefas delegadas por um gerente principal. Normalmente uma base de dados única é utilizada pelo gerente central. Por fim, na arquitetura distribuída existirão diversos gerentes de redes idênticos gerenciando segmentos específicos da rede, comportando-se como gerentes centrais em

cada segmento. Não existe a figura de gerentes intermediários e cada gerente existente possui uma base de dados própria, possivelmente replicada nos outros gerentes da rede.

A arquitetura centralizada é simples de ser implantada porque envolve apenas a instalação do *software* da plataforma de gerência, e agentes nos dispositivos gerenciados. Logo, o modelo proposto é mais complexo de ser instalado que a arquitetura centralizada porque os consumidores de políticas, monitores de QoS e identificadores de alvos precisam também ser instalados. Além disso, várias associações devem ser criadas no ambiente de gerência para que os elementos intermediários sejam atrelados aos alvos correspondentes.

Na arquitetura hierárquica, os gerentes intermediários também precisam ser instalados, e neste sentido a complexidade do modelo proposto é similar, pois na prática os elementos intermediários (consumidores, monitores e identificadores de alvos) são gerentes intermediários com funções mais específicas. A realização de associações na estação de gerência também é necessária na arquitetura hierárquica assim como é no modelo proposto. Resumidamente, as complexidades de instalação neste caso são similares, não muito diferentes entre a arquitetura hierárquica e o modelo proposto.

Por fim, a arquitetura distribuída exige a instalação de todos os gerentes em diversos segmentos. Proporcionalmente, esta atividade leva mais tempo que a instalação de gerentes intermediários, já que os gerentes da arquitetura distribuída são pacotes de *software* completos, enquanto que os gerentes intermediários possuem funções mais específicas, mas menos *software* é necessário. Como os elementos intermediários do modelo são também gerentes intermediários, comparativamente, a instalação de uma arquitetura de gerência distribuída, neste aspecto, é mais complexa pelo tempo gasto. Entretanto, na arquitetura distribuída não existe a necessidade de se registrar gerentes intermediários, já que todos os gerentes funcionalmente operam de forma centralizada em seus segmentos. Logo, neste outro aspecto, a arquitetura distribuída torna-se menos complexa em sua instalação. De forma global, mas por outros motivos, pode-se também dizer que neste caso a complexidade de instalação do modelo proposto, comparado com a instalação de uma arquitetura distribuída, é similar.

Pelos elementos que o modelo apresenta, e pelas necessidades de criação de associações existentes, no geral a instalação de uma solução de gerência baseada no modelo proposto é mais complexa que a instalação de outras soluções de gerência. Entretanto, a complexidade de instalação é enfrentada apenas uma vez (na própria instalação), não se repetindo novamente (exceto quando elementos adicionais são colocados na rede). Além disso, cabe lembrar que a tarefa de implantação de QoS é uma facilidade auxiliar que ajuda na diminuição das complexidades de implantação.

Complexidade de utilização

Depois que um sistema de gerência estiver implantado, o administrador de rede passa a utilizá-lo para gerenciar sua rede. O modelo apresentado na tese propõe o uso de abstrações para implementar facilidades que auxiliem o gerenciamento de rede. Novamente, as arquiteturas centralizada, hierárquica e distribuída serão utilizadas para comparação em relação à complexidade de utilização.

De forma geral, as arquiteturas padrão visam gerenciar uma rede de computadores através de um método padrão de acesso aos dispositivos. Há anos existe

um esforço internacional em tornar o SNMP o método padrão citado. Como conseqüência, a maior parte dos equipamentos de rede atualmente possuem suporte ao protocolo e isso deveria ser suficiente para as arquiteturas. Entretanto, o SNMP não é o único meio de acesso aos dispositivos: outros protocolos podem ser utilizados. Além disso, o conjunto de informações de gerência é diferente para dispositivos diferentes. Como resultado, não existe na prática uma forma comum de lidar com as informações de gerência em uma rede de computadores heterogênea. As arquiteturas de gerenciamento acabam tendo que suportar, ao mesmo tempo, diversas modalidades de acesso aos dispositivos, e tratamentos diferenciados dos mesmos porque as informações obtidas também são diferenciadas. Logo, as então desejadas facilidades de uso das arquiteturas de gerenciamento padrão não são totalmente alcançadas na prática.

Tanto o modelo proposto como as soluções de PBNM, tentam no fundo implementar uma nova camada de abstração entre as estruturas de rede e o sistema de gerência. A nova camada de abstração tem a função de apresentar uma interface de gerência padrão ao sistema de gerenciamento, apesar de lidar diferentemente com os dispositivos. As políticas utilizadas implementam esta camada de abstração, e os elementos intermediários são os responsáveis por executarem as traduções para ações específicas nos dispositivos. Por estas considerações, a simples existência de uma nova camada de abstração que apresenta uma interface padrão de gerência possibilita uma complexidade de utilização do modelo muito menor, quando comparada com a complexidade de utilização de uma arquitetura de gerência padrão.

Apesar da complexidade de utilização comparativamente ser baixa, o modelo pode ainda implementar operações ainda menos complexas. Isso é possível quando o ambiente de gerência implementa abstrações ainda maiores das complexidades da rede. O segundo exemplo da seção 4.11 discutiu como novas abstrações podem ser realizadas no ambiente de gerência.

O conjunto de abstrações existentes é dependente de implementação, mas novamente, o modelo apresenta uma complexidade menor que as arquiteturas tradicionais apenas por apresentar uma primeira abstração das complexidades da rede. Outras abstrações podem ser implementadas no ambiente de gerência em implementações específicas.

Complexidade de suporte por parte dos fornecedores de equipamentos de rede

Atualmente, a tendência no suporte à gerência de rede nos equipamentos é a implantação de agentes SNMP. Quase todo novo equipamento produzido fornece um agente SNMP. A versão 1 do protocolo ainda é a mais encontrada, mas diversos fornecedores começam a suportar fortemente o SNMPv3.

Os trabalhos de pesquisa desenvolvidos em universidades e principalmente o apoio do IETF em relação ao PBNM têm demonstrado que a existência de suporte PBNM internamente aos equipamentos pode ser uma realidade. E tal suporte mostra-se como a implementação de consumidores de políticas internos aos equipamentos. Ainda pesquisas devem ser desenvolvidas e normas padronizadas, mas a tendência é de que políticas possam realmente ser implantadas diretamente nos equipamentos.

Em relação ao modelo proposto, a maior dificuldade é encontrar suporte à tarefa de monitoração de QoS. As soluções que mais se aproximam do desejado são capazes

apenas de monitorar a vazão dos fluxos (por exemplo, RMON2). Isso acaba forçando a utilização de soluções externas aos equipamentos para que o QoS oferecido possa ser medido. Soluções externas, entretanto, tendem a serem menos precisas, e isso obviamente não é interessante. Apesar do IETF possuir definições para monitoração [BRO 97], não parece ser uma tendência o suporte interno da solução nos equipamentos.

5.5 Escalabilidade do modelo

A escalabilidade do modelo proposto está diretamente relacionada com a possibilidade de distribuição de seus elementos na rede gerenciada. Um dos requisitos do modelo era que o mesmo deveria ser capaz de gerenciar, sem muitas modificações, redes de tamanhos e complexidades diferentes. O ponto principal para a escalabilidade do modelo, neste contexto, é a existência dos elementos intermediários e o modo como os mesmos atuam. Os itens a seguir discutem a questão de escalabilidade em relação a alguns aspectos importantes.

Escalabilidade da descoberta de QoS

A descoberta de QoS confia nos serviços dos identificadores de alvos para a verificação de quais são as facilidades de QoS existentes na rede. O número de identificadores de alvos deve ser proporcional à rede gerenciada. A localização de tais identificadores, por outro lado, deve ser adequada à topologia existente.

Em redes pequenas, em torno de 400 a 500 dispositivos, espera-se que os identificadores de alvos sejam em menor número. Em certos casos os identificadores podem até mesmo não ser usados se o administrador da rede tiver condições de registrar na base de associações todos os alvos, o tipo de cada alvo e a correspondência entre alvos e dispositivos. Na maior parte dos casos, entretanto, um único identificador de alvos pode ser suficiente para uma rede pequena. Este único identificador poderia estar localizado em uma máquina separada, ou junto ao ambiente de gerência. A segunda opção é preferível para evitar o consumo de recursos de rede na comunicação entre ambiente de gerência e identificador.

Em redes maiores, com mais de 500 dispositivos, o número de identificadores pode ser maior. O aumento de identificadores é necessário para que a descoberta de todos os serviços existentes seja realizada de forma mais rápida. Por outro lado, quando se analisa os objetivos da descoberta de QoS e o momento em que ela deve ser realizada, o número de identificadores em redes maiores pode variar bastante. Se a descoberta é principalmente utilizada para auxiliar na implantação de QoS, então o número de identificadores pode permanecer baixo porque neste momento uma descoberta rápida de serviços com QoS não é uma tarefa crítica. O administrador da rede pode esperar pelo processo de descoberta para então definir como a rede deve ser utilizada na implantação de QoS. Nesta situação, independentemente do tamanho da rede gerenciada, a descoberta de QoS é escalável porque o tempo decorrido da descoberta não é crítico. Por outro lado, quando a descoberta de QoS é utilizada como suporte à manutenção de QoS, um processo de identificação de alvos e facilidades de QoS deve ser mais rápido. O extremo crítico dessa situação é quando a descoberta de QoS é utilizada para a identificação "*plug-and-play*" de dispositivos com QoS recentemente colocar na rede. Neste último caso, identificadores de QoS deveriam ser

espalhados na rede, preferencialmente em cada segmento de rede, para que a descoberta fosse agilizada. Considerando que cada segmento possuirá, nestas condições, um identificador de alvos, independentemente do tamanho da rede a identificação ocorrerá de forma adequada, o que significa que a descoberta de QoS, quando uma descoberta rápida for essencial, é escalável em relação ao tamanho da rede também nesta situação.

Outro motivo para a utilização de vários identificadores de alvos é quando, por questões de segurança, os alvos que fazem parte de um segmento protegido não conseguem ser descoberto por identificadores localizados fora do segmento onde os alvos se encontram. Esta situação é típica quando os segmentos estão protegidos por *firewalls* que impedem a passagem de protocolos utilizados na descoberta (por exemplo, SNMP). O uso de um identificar interno ao segmento de interesse resolve o problema, desde que a comunicação com o identificar de alvos seja autorizada no *firewall*.

Escalabilidade da implantação de políticas

Como visto, a implantação de políticas depende da existência de consumidores de políticas que traduzem as mesmas para ações específicas nos dispositivos que contém os alvos de interesse. O número de consumidores de políticas utilizados em uma rede depende dos tipos de alvos existentes, dos protocolos de comunicação com os dispositivos com contém estes alvos, e do tamanho da rede gerenciada.

Os tipos de alvos existentes determinam as ações necessárias para a implantação das políticas. Por exemplo, uma política de priorização de tráfego normalmente está associada a dois tipos de alvos: as disciplinas de filas e os processos de classificação de pacotes. Cada alvo de um determinado tipo normalmente possui um consumidor de políticas para aquele tipo de alvo. Assim, existirá um consumidor de políticas para as disciplinas de filas e outro consumidor de políticas para os processos de classificação de pacotes. Na prática, os dois tipos diferentes de consumidores podem ser implementados em uma mesma entidade, comportando-se como um único consumidor, mas que funcionalmente são dois. Como resultado, quanto maior a diversidade de tipos de alvos existentes em uma rede, maior será o número de consumidores de políticas existentes. Em uma rede com poucos tipos (por exemplo, uma rede que possui apenas disciplinas de filas), o número de consumidores será menor.

Para simplificação da análise, supõe-se agora uma rede com um único tipo de alvo: disciplinas de filas. Nesta rede, o número de consumidores pode também variar, de acordo com o conjunto de protocolos de comunicação necessários para se acessar os dispositivos quem contém os alvos. Em uma mesma rede, um roteador de um fornecedor A pode usar SNMP para a programação das filas, enquanto que outro roteador B pode usar Telnet para tal. Nesta situação, dois consumidores diferentes são necessários: um para implantar políticas nas disciplinas de filas do roteador A via SNMP, e outro para implantar políticas nas disciplinas de filas do roteador B via Telnet. Se existir um terceiro roteador que necessita ainda outro protocolo de comunicação, então mais um consumidor de políticas será necessário. Como conseqüência, quanto maior o número de protocolos maior o número de consumidores necessários.

Apesar de aparentemente grande, o número de consumidores de políticas existentes pode ser diminuído porque um mesmo consumidor pode programar vários alvos do mesmo tipo cujos equipamentos utilizam o mesmo protocolo de acesso. Por exemplo, em uma rede com equipamentos de um mesmo fornecedor, que por padrão

utiliza SNMP para programar as disciplinas de filas, um único consumidor de políticas é suficiente. Neste caso, o único consumidor irá acessar todas as disciplinas (tipo do alvo) utilizando SNMP (o protocolo de acesso). Por outro lado, mesmo em uma rede mais "padronizada" como a do exemplo, outros consumidores podem ser necessários devido à forma como é apresentada a topologia da rede. Se a rede possuir muitos segmentos localizados fisicamente distantes ou que possuem banda de acesso disponível limitada, é adequado se colocar consumidores de políticas nos segmentos remotos. Logo, com um número grande de equipamentos (e como conseqüência, um número grande de segmentos) espera-se que o número de consumidores de políticas também seja maior.

A utilização de vários consumidores de políticas acaba sendo uma tendência porque as redes de computadores são naturalmente heterogêneas, possuindo diversos tipos de alvos e protocolos de comunicação com os dispositivos. Soma-se a isso também as dimensões das redes atuais. Neste contexto, a escalabilidade é uma questão importante porque o volume de tráfego de gerência trocado entre o ambiente de gerência e os consumidores de políticas pode ser considerável. Além desse tráfego, existirá ainda o tráfego necessário para a transferência de políticas da base de políticas até os consumidores, quando se está utilizando o modelo *pull* de transferência. Levando-se em conta apenas o tráfego de notificações (entre ambiente de gerência e consumidores, e vice-versa), pouco provavelmente a rede seja impactada, já que mensagens de notificação consomem muito pouca banda. A questão principal acaba sendo a transferência de políticas aos consumidores, e neste ponto o modelo de transferência adotado é importante em relação à escalabilidade do modelo. Inicialmente, assumindo que a transferência será feita via modelo *push*, a política é transferida da estação de gerência principal até os consumidores diretamente, sem existir interação do consumidor com o repositório de políticas. Neste caso, o tráfego de rede nas proximidades do ambiente de gerência é proporcional ao número de consumidores existentes. Em redes heterogêneas e grandes, o tráfego próximo à estação de gerência pode até mesmo causar congestionamentos, pondo em risco a qualidade da gerência da rede. Entretanto, o tráfego de gerência, comparativamente com o tráfego de gerência gerado por plataformas padrão, tende a ser menor porque as políticas transferidas aos consumidores permanecem neste consumidor para serem executadas nos momentos especificados. Isso permite ao administrador fazer uma transferência de políticas espalhada no tempo, distribuindo a carga gerada. Isso permite também que momentos de baixa utilização da rede possam ser utilizados para fazer a transferência de um número maior de políticas, sem prejudicar o andamento dos outros serviços.

No modelo *pull* a questão da carga de gerência próximo ao ambiente de gerência é menos crítica. Neste modelo o principal tráfego gerado é aquele entre os consumidores de políticas e a base de políticas. Assim, o tráfego entre ambiente de gerência e consumidores de políticas é resumido ao tráfego de notificações, que como visto, não é crítico. Felizmente, a base de políticas não necessariamente precisa estar localizada no mesmo dispositivo que implementa o ambiente de gerência. Além disso, pode ainda utilizar diversas bases de políticas diferentes ao mesmo tempo. Se cada base possível for colocada próxima aos consumidores que utilizarão as políticas armazenadas, o tráfego de gerência total acaba sendo distribuído em vários segmentos, sem a centralização que era encontrada no modelo *push*. O administrador da rede cria políticas que podem ser armazenadas em bases distantes. No momento da implantação de uma política, a notificação gerada pelo ambiente descreve, possivelmente através de um URL, qual a política que o consumidor deve utilizar e qual a base que deve ser acessada para que a

política seja recuperada.

Comparativamente, o modelo *push* é mais adequado quando a rede gerenciada possui um número de alvos reduzido e a base de políticas é única e localizada no mesmo dispositivo que contém o ambiente de gerência. Já o modelo *pull* é mais adequado para a gerência de redes com um número maior de consumidores. Neste caso, espera-se que existam várias bases de políticas espalhadas na rede, mais "próximas" dos consumidores usuários das políticas.

Escalabilidade da monitoração de QoS

Em termos gerais, as questões de escalabilidade da monitoração de QoS são similares às questões de escalabilidade da implantação de políticas. Entretanto, na monitoração de QoS as questões acabam sendo menos críticas porque, proporcionalmente, o número de monitores de QoS ativos na rede é menor que o número de consumidores de políticas. Além disso, o número de políticas monitoradas é menor que o número de políticas implantadas. Isso é decorrência do fato de que nem toda política implantada na rede é uma política também monitorada. Apenas as políticas mais críticas são verificadas, o que faz com que os monitores de QoS sejam menos utilizados que os consumidores de políticas.

Da mesma forma que os consumidores, o modelo define que cada alvo também possuirá um monitor de QoS associado. O número de monitores utilizados na rede, neste caso, também é dependente do número de tipos de alvos existentes, da variedade de protocolos de comunicação com os dispositivos e das dimensões da rede gerenciada. Outro fator que influencia no número de monitores é a rota de rede utilizada para o transporte de dados em uma sessão. Por exemplo, em uma aplicação de videoconferência onde a rota entre origem e destino é composta por roteadores de um mesmo fornecedor, com método de acesso idêntico, possivelmente um único consumidor de políticas é necessário para a programação das filas nas interfaces dos roteadores. Supondo que o número de roteadores existe é cinco, o número de monitores de QoS necessários para a verificação do QoS alcançado com a política implantada necessariamente será maior que um, e pode ser inclusive cinco (um monitor de QoS para cada roteador). Assim, neste aspecto, o número de monitores necessários acaba sendo maior que o número de consumidores de políticas.

Aqui, uma distinção deve ser feita entre o número de monitores implantados e o número de monitores utilizados. Pelo último exemplo, conclui-se que o número de monitores implantados na rede tende a ser maior que o número de consumidores de políticas porque em uma rota de uma sessão vários monitores podem ser necessários para monitorar uma política implantada por um único consumidor. Entretanto, nem todos os monitores implantados na rede são utilizados porque nem todas as políticas devem ser observadas. Assim, o número de monitores implantados é maior que o número de consumidores, mas o número de monitores utilizados tende a ser menor porque nem todas os políticas devem ser monitoradas. Conseqüentemente, a solução acaba sendo também escalável em relação à monitoração de QoS e as mesmas considerações sobre os consumidores de políticas em relação à transferência de políticas são também aplicadas aos monitores de QoS. Para redes pequenas, com um menor número de monitores o uso de uma base centralizada junto ao ambiente de gerência e que utiliza o modelo *push* é recomendada. Para redes maiores, com um maior número de

monitores, bases distribuídas e operando no modelo *pull* devem ser utilizadas.

Escalabilidade e bases de dados

Indiretamente os itens anteriores discutiram sobre questões de escalabilidade relacionadas com as bases de dados. O ponto principal a ser verificado é a necessidade de se ter bases distribuídas de dados em relação ao número de dispositivos existentes na rede gerenciada. Nas discussões sobre as escalabilidades da implantação de políticas e da monitoração de QoS, ficou claro que quanto maior a rede, maior a tendência em se ter uma base de políticas distribuída nos vários segmentos e operando no modelo *pull*. Esse requisito era necessário para que o tráfego de gerência de rede não ficasse concentrado nas proximidades do ambiente de gerência. O problema da concentração de tráfego de gerência, em relação à transferência de políticas, é resolvido com a simples alteração da localização da base de políticas. Entretanto, uma realocação na prática só iria desviar o tráfego de políticas do segmento do ambiente de gerência para o novo segmento onde se encontraria a base de políticas. Para uma distribuição desse tráfego, então surge a necessidade também da distribuição dessa base. Colocando porções da base de políticas "perto" dos consumidores de políticas e monitores de QoS de interesse acaba-se por confinar o tráfego em segmentos específicos, não influindo em outros segmentos de rede.

Considerando que o tráfego de gerência gerado pelas notificações é baixo, comparado com os outros tráfegos, pode-se afirmar que a base de *status* é a menos afetada pelas questões de escalabilidade. Os fatores que mais influenciam a base de *status* são as dimensões da rede gerenciada, o número de monitores de QoS implantados e o número de políticas monitoradas. Com um número maior de dispositivos na rede o número de monitores utilizados é maior. Com vários dispositivos é também maior o conjunto de políticas utilizadas e como conseqüência o número de monitores utilizados aumenta. Assim, o número de notificações proporcionalmente também é maior. Um quarto fator importante é também a qualidade da arquitetura de fornecimento de QoS utilizada. Uma arquitetura de melhor qualidade irá permitir que os serviços recebam os tratamentos adequados definidos nas políticas e as degradações serão em menor número. Isso faz com que, apesar do eventual grande número de monitores de QoS utilizados, o número de notificações seja pequeno, já que menos degradações serão notificadas. A base de *status* pode começar a sofrer interferências mais sérias quando a arquitetura de QoS implantada não possuir uma boa qualidade em uma rede de grandes dimensões. Neste caso, o número de notificações geradas poderá levar o congestionamento do segmento onde a base de *status* se encontra (normalmente junto ao ambiente de gerência). Nestas situações a distribuição da base passa a ser uma solução importante.

Funcionalmente, a distribuição da base de *status* poderia ser interessante para acompanhar a distribuição da base de políticas. Assim, as duas bases poderiam ser distribuídas uniformemente na rede, até mesmo sendo implementadas nos mesmos dispositivos nos segmentos existentes. Dessa forma, políticas e seus estados internos seriam informações sempre localizadas "próximas" umas das outras. Entretanto, uma interação crítica com a base de *status* pode ser prejudica com a distribuição: a comunicação entre bases de *status* a ambiente de gerência. Diferentemente do que acontece com a base de políticas, a base de *status* é bem mais consultada pelo administrador da rede ou por processos automatizados dentro do ambiente de gerência. A análise de QoS, como visto, é também um grande cliente da base de *status* e da

monitoração de QoS. Como a interação do ambiente de gerência com a base de *status* tende a ser mais volumosa que as notificações geradas pelos monitores de QoS, a distribuição da base de *status* só é possível quando existirem recursos de rede suficientes para a comunicação entre ambiente de gerência com as diversas bases existentes. Caso contrário, o local mais apropriado para base de *status* é realmente junto ao ambiente de gerência.

Por fim, a base que apresenta os maiores problemas em relação à escalabilidade é a base de associações. Seguramente a base de associações é a base que mais recebe consultas (escritas, remoções ou leituras) do ambiente de gerência. É a base que possui todas as associações existentes entre os elementos do modelo, e como tal é essencial para todos os processos existentes junto ao ambiente de gerência. A interação dos elementos intermediários com a base de associações, por outro lado, é a menor de todas. Tirando os identificadores de alvos, que operam apenas em momentos específicos definidos pelo administrador, os outros elementos intermediários não interagem diretamente com a base de associações. Isso faz com que a mesma seja mais independente das dimensões da rede em relação às comunicações. Isto é, mesmo que o número de equipamentos da rede seja extremamente grande, o acesso à base de associações não sofre nenhum problema porque o único elemento intermediário que se comunica com a mesma são os identificadores de alvos. Por outro lado, o tamanho da base de associações é naturalmente proporcional ao tamanho da rede gerenciada. Com uma rede maior, mais alvos irão existir, mais consumidores e monitores associados precisarão ser definidos e um número maior de identificadores de alvo poderá ser utilizado. O tamanho da base é também proporcional ao número de políticas em uso na rede, porque uma política em uso gera atualizações para as associações da política com os alvos onde a mesma deve ser implantada. Apesar disso, o tamanho da base não é proporcional ao número de políticas definidas, porque uma política definida não obrigatoriamente precisa ser implantada, e assim nenhuma associação é gerada para essa política.

Como regra geral, a possibilidade de distribuição da base de associações é pouco provável. Inicialmente, porque o tamanho da rede não afeta a comunicação com a base (apesar de afetar o seu tamanho), e também porque as interações entre a base de associações e ambiente de gerência de rede é tão grande que a realocação da base para outro segmente faria com que os processos dependentes da base (localizados junto ao ambiente de gerência) passassem a operar mais lentamente (já que a comunicação imporia um atraso maior nas transações), além de potencialmente introduzirem muito tráfego desnecessário em segmentos de rede para a comunicação com a base. Assim, desde que a base de dados permaneça junto ao ambiente de gerência, o modelo proposto também é escalável em relação à base de associações.

Escalabilidade e o ambiente de gerência

Enquanto os elementos intermediários do modelo possuem uma certa predisposição para a distribuição, o ambiente de gerência e suas estruturas internas (processos para visualização, análise e implantação de QoS) mostram-se como os elementos de características mais centralizadas do modelo.

De forma geral, a centralização do ambiente de gerência é uma característica importante para um administrador de rede porque permite a ele acessar todas as

facilidades de gerência a partir de um único ponto comum. Isso elimina complexidades inerentes de um sistema de gerência totalmente distribuído facilitando às tarefas que devem ser executadas. A centralização também permite que informações relacionadas possam ser mais bem analisadas, pois se encontram disponíveis (ainda que processos de coleta remota tenham que ser utilizados) em um mesmo ambiente.

Os estudos para a criação de estruturas que permitam um gerenciamento distribuído buscavam soluções principalmente para o alto tráfego de gerência gerado junto às estações de gerência, além da delegação de tarefas a agentes inteligentes (ou gerentes intermediários) autônomos que implementavam tarefas de gerência mesmo quando o administrador da rede não estava ativamente influenciando no ambiente. Sob estes aspectos, o modelo proposto é adequado. A descentralização das tarefas de gerência é alcançada naturalmente com os elementos intermediários do modelo, que na prática são gerentes intermediários especializados. A proposta da utilização de Script MIB para comunicação com tais elementos no ambiente QAME citado anteriormente é mais uma mostra do real papel de gerentes que os elementos intermediários do modelo executam. Apesar desta descentralização encontrada, o ambiente apresenta uma interface única de gerência ao administrador por possuir um ambiente de gerência centralizado. Internamente, os processos existentes, por não diretamente interagirem muito com a rede gerenciada, não são afetados pela centralização.

Eventualmente, um ambiente de gerência descentralizado poderia ser utilizado caso diversos administradores precisassem ter acesso simultâneo ao ambiente. Felizmente, com a utilização das tecnologias Internet e do gerenciamento baseado na Web, um único ambiente de gerência pode ser adequadamente utilizado ao mesmo tempo por administradores localizados distantes fisicamente uns dos outros, muitas vezes distantes fisicamente até mesmo da rede gerenciada.

Ainda assim, uma rede poderia ser adequadamente gerenciada por mais de um ambiente de gerência ao mesmo tempo. Neste caso a descentralização dos ambientes poderia levar a uma situação onde o gerenciamento se tornaria complexo. Entretanto, como visto anteriormente, um dos principais integradores das tarefas de gerência são as bases de dados. Como as bases não fazem parte do ambiente de gerência, vários ambientes diferentes poderiam utilizar as mesmas bases de dados e assim ter acesso comum ao mesmo conjunto de informações. A parte mais crítica neste modelo de operação relaciona-se com a base de associações, que como visto anteriormente, opera mais adequadamente quando localizada junto ao ambiente de gerência. Num cenário onde vários ambientes acessam a mesma base, uma questão crítica é determinar onde tal base deve estar localizada. Independentemente de sua localização, um tráfego maior de rede será observado porque é impossível a base ser local a todos os ambientes de gerência existentes. Logo, esta última observação indica que a centralização do ambiente de gerência, além de ser interessante para o gerente de rede, é também aconselhada pelo problema relacionado à base de associações.

Em relação ao tamanho da rede gerenciada, o ambiente de gerência não é afetado profundamente. Com uma rede maior, a distribuição é realizada apenas para os elementos mais críticos que interagem diretamente com a rede (consumidores de políticas, monitores de QoS e identificadores de alvos). O ambiente de gerência, mesmo em redes maiores, não precisa ser distribuído porque seus elementos internos, que pouco interagem comparativamente com a rede, operam adequadamente. Em redes maiores, entretanto, a existência de vários administradores é uma necessidade, mas mesmo neste

caso, com o uso de tecnologia Web, o acesso adequado e simultâneo ao ambiente é conseguido. Isso leva a crer que não apenas a existência de um ambiente centralizado é importante para o gerente da rede, mas como é aconselhado para o adequado funcionamento do modelo, já que a escalabilidade do mesmo em relação ao ambiente de gerência é adequada.

5.6 Exeqüibilidade do modelo através do protótipo QAME

Nas seções anteriores foi possível verificar as principais características do modelo. Além das anteriores, a exeqüibilidade é um fator importante porque indica que o modelo pode ser utilizado na prática para a gerência de um rede com QoS. Nesta seção é apresentada a terceira importante contribuição da tese: o ambiente de gerência QAME (*QoS-Aware Management Environment*) [GRA 2001g], um protótipo implementado que segue as definições do modelo de gerência proposto.

A arquitetura do QAME já havia sido anteriormente apresentada na seção 5.2. Como mostrado, o ambiente implementa os três níveis de elementos do modelo (elementos inferiores, intermediários e superiores). O protótipo foi testado em uma rede com priorização de tráfego baseada em roteadores com sistema operacional Linux. A FIGURA 5.3 apresenta o esquema geral da rede de teste.

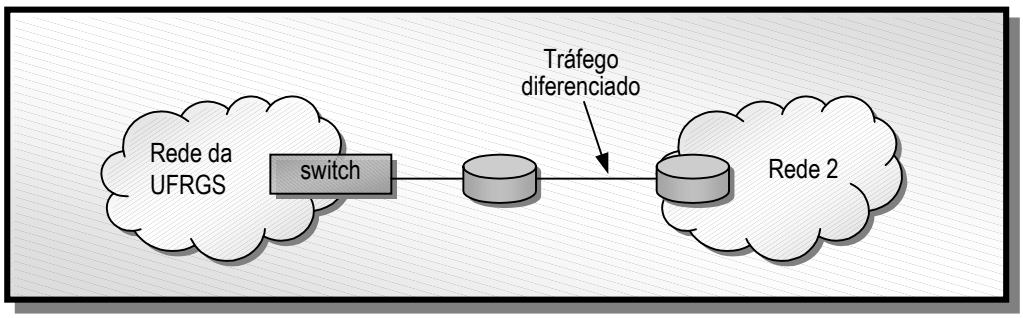

FIGURA 5.3 – Ambiente de testes do protótipo QAME

Nesse ambiente, os alvos da gerência eram as várias filas implementadas em cada interface dos roteadores, bem como seus processos de classificação de tráfego. Um consumidor de políticas associado aos alvos dos roteadores foi desenvolvido e colocado externamente a cada roteador. Um monitor de QoS, também externo aos roteadores, controla a verificação de fluxos executada internamente por agentes coletores. Os agentes são capazes de verificar degradações de *jitter* e vazão, enquanto o monitor externo detecta problemas de perda e atraso dos fluxos. O anexo1 apresenta uma descrição mais detalhada dos elementos.

O ambiente de gerência foi totalmente implementado em uma máquina Linux utilizando tecnologia PHP4 [PHP 2001]. O acesso do administrador ao ambiente se dá via HTTP/HTTPS, implementado assim uma gerência de QoS baseada na Web [GRA 2001a]. A visualização de QoS é implementada com tecnologia Flash [MAC 2001a], que permite uma interação mais dinâmica do administrador da rede através de seu navegador Web. Normalmente, os ambientes de gerência baseados na Web não permitem uma interação muito grande porque a maior parte das imagens disponibilizadas são geradas na forma de bitmaps estático. Com o uso de Flash,

entretanto, o administrador pode, por exemplo, alterar a disposição dos equipamentos nos mapas de rede, ter acesso a menus sensíveis ao contexto, definir enlaces entre dispositivos através do desenho de linhas, entre outros.

O servidor Web é responsável por receber requisições HTTP/HTTPS do cliente (navegador Web) e proceder com consultas aos bancos de dados ou com consultas diretas aos dispositivos gerenciados. A consulta aos bancos de dados é realizada utilizando-se *scripts* PHP4.

As bibliotecas SNMP são necessárias para permitir o acesso direto aos dispositivos gerenciados pelo sistema. Foram utilizadas as bibliotecas NET-SNMP [UNI 2001] por serem de código aberto e já suportadas no PHP4. Para que os serviços com QoS possam ser gerenciados, os equipamentos de rede devem possuir agentes SNMP que implementam MIBs (*Management Information Base*) QoS. Ainda que não existam MIBs QoS disponíveis, o QAME pode gerenciar os dispositivos acessando MIBs padrão como MIB-II, RMON2, entre outras.

O navegador Web utilizado deve possuir suporte à tecnologia Flash versão 5.0 [MAC 2001a]. Esta tecnologia permite ao gerente interagir de uma forma completamente dinâmica em relação aos gráficos da topologia da rede gerenciada. Além disso, o uso de Flash reduz muito o tráfego de informações entre o navegador Web e o servidor, já que a apresentação da topologia requer atualmente a transferência de apenas 20 Kbytes de código binário ao navegador. Neste contexto, o gerente pode realizar as seguintes ações:

- Navegar na topologia de rede;
- Adicionar e remover dispositivos;
- Adicionar e remover ligações físicas ou lógicas entre dispositivos;
- Editar a disposição gráfica dos elementos da topologia.

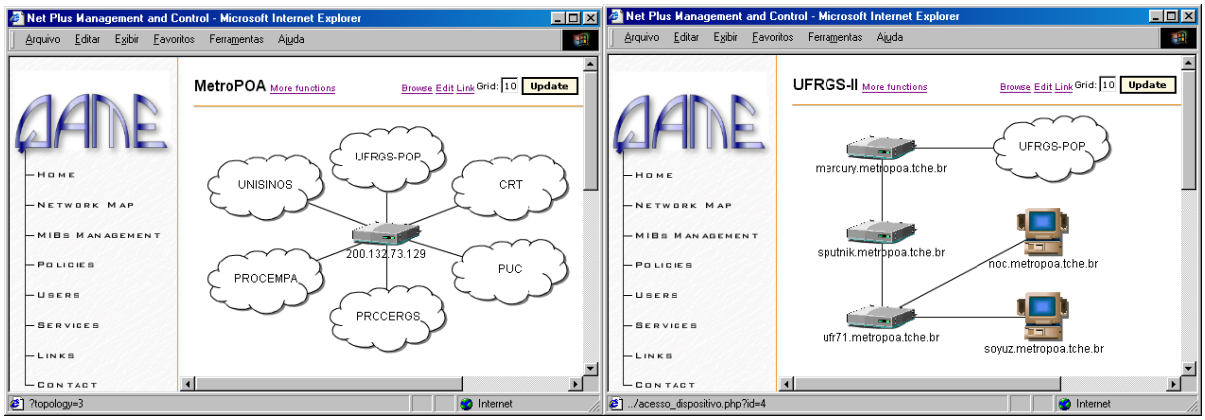

FIGURA 5.4 – Interface gráfica QAME

A FIGURA 5.4 apresenta a interface gráfica baseada na Web do QAME, mostrando a topologia principal da rede do MetroPOA. Apresenta também a sub-rede do instituto de informática da UFRGS, acessada a partir da navegação iniciada pela topologia principal (à esquerda). O sistema opera em três modos distintos: navegação, edição e ligação. No modo navegação o usuário percorre a rede gerenciada acessando as várias sub-redes existentes. No modo de edição é permitido se alterar a disposição gráfica dos elementos e remover elementos e ligações. Por fim, no modo ligação é

possível se criar ligações entre os elementos de uma topologia.

As bases de dados foram implementadas utilizando o banco MySQL [YAR 99] e o serviço de diretórios OpenLDAP [UNI 96]. O MySQL implementa a base de associações e a base de *status*, enquanto o OpenLDAP implementa a base de políticas. O MySQL foi utilizado por ser rápido e não consumir muitos recursos de máquina, se comparado aos bancos tradicionais. O OpenLDAP foi escolhido por ser de código aberto e por ser bem conceituado em relação às implementações de LDAP. Ambos MySQL e OpenLDAP possuem suporte nativo no PHP4, o que facilitou o desenvolvimento do sistema de gerência pois as tecnologias necessárias já possuíam integração entre si.

A comunicação com as bases de dados, a partir dos elementos intermediários, se dá através de LDAP, mensagens *InformRequest* e HTTP. Para tal, interfaces extras foram desenvolvidas. O acesso à base de políticas foi conseguido pela API de programação disponibilizada com o pacote OpenLDAP. O suporte às mensagens *InformRequest* na base de políticas foi alcançado com o uso da solução NET-SNMP. Por fim, um conjunto de *scripts* PHP4 possibilitou o acesso, via HTTP, da base de associações a partir dos elementos intermediários. A comunicação com os elementos intermediários, via Script MIB, é possível graças a utilização do *software* Jasmin [TU 99], desenvolvido pela Universidade de Braunschweig.

A definição de políticas no ambiente é feita através de um conjunto simples de formulários. As políticas são definidas através de identificação de portas origem e destino, endereço de rede origem e destino, protocolo de transporte, e uma associação com uma classe de serviços. Outros formulários são utilizados para criar, remover e editar as classes de serviços existentes.

A descoberta de QoS é suportada através de um módulo localizado junto ao ambiente de gerência que investiga a rede, descobre máquinas ativas, e determina os serviços com suporte à QoS existentes através da consulta a MIBs SNMP. Foram implementadas, nos dispositivos (roteadores Linux), MIBs simples capazes de informar a existência das facilidades de priorização de tráfego existentes. Nos sistemas finais, agentes SNMP também foram criados para executarem a marcação de tráfego em serviços diferenciados [GRA 2000b].

A análise de QoS é suportada no ambiente pela utilização do *software* MRTG [OET 2001]. Originalmente criado para a monitoração da utilização de enlaces em roteadores, o MRTG é suficientemente flexível para traçar gráficos sobre quaisquer informações disponibilizadas. Para tal, um método de consulta deve ser fornecido (o método padrão é o uso de SNMP nativo), e as identificações das informações a serem coletadas devem ser definidas. A visualização da análise de QoS é imediata porque o MRTG gera páginas HTML disponibilizadas ao servidor Web. A integração do MRTG com o resto dos componentes é conseguida porque referências às páginas criadas pelo MRTG estão disponíveis em todo o ambiente. Por exemplo, um administrador interessado em monitorar os enlaces de um roteador de borda de seu ambiente administrativo deve inicialmente navegar na topologia de rede para acessar o roteador de interesse. Logo em seguida, uma lista de interfaces do roteador é apresentada, e o administrador escolhe quais as interfaces de interesse devem ser monitoradas. Tal escolha gera um conjunto de definições de configuração MRTG que permitem que o *software* passe a monitorar as interfaces escolhidas. Todo o dispositivo que tiver pelo menos uma interface sendo monitorada terá, no ambiente, uma referência às páginas

HTML geradas pelo MRTG que descrevem a utilização da interface. O administrador pode ainda selecionar os dispositivos de interesse através de um mecanismo de busca existentes na visualização de QoS. O mecanismo recebe como entrada o endereço de rede ou nome do dispositivo e procura, na base de associações, pela ocorrência do mesmo. Quando uma ocorrência é encontrada, a topologia a que o dispositivo faz parte é determinada, e um mapa da topologia gerado dinamicamente. Neste mapa, o dispositivo procurado é apresentado de forma destacada dos demais dispositivos da mesma topologia.

O desenvolvimento do protótipo foi realizado durante dois anos. Neste processo, pode-se concluir, sobre a exeqüibilidade do modelo, os seguintes itens:

- A construção de um ambiente de gerência que siga o modelo proposto foi complexa porque o modelo e o ambiente estavam sendo desenvolvidos em paralelo. Muitas definições necessárias à implementação ainda estavam sendo criadas no modelo, fazendo com que o desenvolvimento tomasse um ritmo lento por vezes. Por outro lado, o desenvolvimento em paralelo permitiu que muitos conceitos do modelo fossem refinados de acordo com as questões práticas levantadas no desenvolvimento. Partindo do pressuposto que o modelo de gerência encontra-se já definido, a complexidade da implementação seria bem menor.
- O ponto mais crítico na implementação do ambiente foi a inexistência de um padrão claro para a definição de políticas e de operação de um sistema que suporte as mesmas. O conceito de gerência baseada em políticas é bem conhecido, porém sua efetivação ainda não é totalmente clara. Existem poucos protótipos de sistemas desenvolvidos, e os produtos comerciais apresentam políticas como regras sem uma abstração muito alta. A falta de definição de padrões para o armazenamento de dados também foi crítica, o que na prática motivou as definições de três bases de dados diferentes no modelo. A implementação de suporte a políticas desenvolvido foi própria da implementação, como dito, por não existir um padrão definido para esses aspectos.
- A integração das tarefas de gerência em um único sistema de gerência que siga o modelo é possível, e não necessariamente complexa. Diversas ferramentas padrão foram utilizadas (por exemplo, MRTG e NET-SNMP). A integração é principalmente facilitada pela existência de bases de dados comuns. A programação de processos que coordenem as diversas tarefas também não é complexa porque as soluções internamente utilizadas eram muito parametrizáveis, permitindo uma comunicação facilitada entre os elementos. Por exemplo, a criação de uma nova entrada na base de *status* (MySQL) para um política (LDAP) associada a um alvo da base de associações (MySQL) foi rapidamente desenvolvida pela existência de suporte à tais bases no PHP4.

Detalhes complementares sobre o ambiente QAME estão presentes nos artigos incluídos no anexo 2 desta tese. Apesar dos artigos apresentarem a implementação do ambiente e suas características, o ponto mais importante a ser verificado é que o ambiente QAME indica que o modelo de gerência de QoS proposto é exeqüível, mesmo para o gerenciamento de redes complexas, como é o caso de redes IP com suporte a QoS.

6 Conclusões

O gerenciamento de redes que possuem QoS é uma tarefa complexa porque redes com QoS são redes mais complexas. O conjunto de soluções de gerenciamento de QoS existente é diversificado, e cada solução ataca aspectos diferentes do QoS. Isso pode ser visto nas pesquisas existentes e mesmo dentro dos trabalhos do IETF. A principal conseqüência desta situação é a inexistência de uma solução para gerência integrada de redes com QoS. Este foi então o ponto principal desta tese: a criação de definições que busquem uma gerência integrada de QoS.

A tentativa de organização e classificação das pesquisas que envolvem a gerência de QoS resultou na primeira contribuição da tese: a definição de seis tarefas de gerência de QoS. Sob estas seis tarefas as pesquisas puderam ser classificadas em implantação, descoberta, manutenção, monitoração, análise e visualização de QoS. Duas principais conclusões podem ser tiradas da análise da classificação realizada. A primeira é de que na prática, utilizando soluções de mercado ou acadêmicas, todas as tarefas de gerência definidas podem ser realizadas, permitindo ao administrador da rede proceder em sua gerência com as seis tarefas definidas. Logo, a classificação acaba por auxiliar o administrador na organização e sistematização da gerência. Como as tarefas são relacionadas umas com as outras, é possível a um administrador, de posse da classificação, organizar a troca de informações entre as tarefas para que a gerência da rede em relação ao QoS ocorra de forma adequada. A segunda conclusão importante é a percepção de que, se as tarefas podem ser na prática executadas, isso acontece de forma não integrada, e um modelo de integração passa a ser necessário. O principal benefício da integração das tarefas é fazer que o administrador não tenha que proceder com diversas definições relacionadas a um mesmo aspecto de gerência para cada ferramenta utilizada. Como não existe integração, cada ferramenta necessita de definições próprias. Com a integração, entretanto, todas as ferramentas utilizadas compartilhariam um único conjunto de definições. Isso diminuiria o tempo gasto nas definições, além de agilizar a troca de informações entre ferramentas.

Um modelo para gerência integrada da QoS foi proposto. O modelo visa a integração das seis tarefas de gerência definidas em um ambiente único de gerência de forma a possibilitar uma comunicação entre os elementos que implementam as tarefas. O modelo foi definido a partir de trabalhos do IETF, mas concentrando-se no gerenciamento de QoS. Com o uso de políticas, o administrador é capaz de determinar como uma rede com QoS deve se comportar. Além disso, com as mesmas políticas, o administrador pode verificar se as definições de comportamento estão realmente sendo cumpridas pelos mecanismos de fornecimento de QoS. Os identificadores de alvos do modelo são ainda responsáveis por vasculhar a rede gerenciada à procura de suporte à QoS nos dispositivos da rede, e processos de análise e implantação complementam o modelo, que finalmente utiliza processos de visualização para permitir a interação entre o ambiente de gerência e o administrador da rede. O modelo proposto foi ainda analisado de onde pode-se concluir os seguintes itens:

- O modelo é capaz de executar a gerência integrada de QoS porque os elementos do modelo fornecem suporte às seis tarefas de gerência de QoS definidas. Além disso, como existe comunicação entre as várias atividades executadas pelos elementos, o suporte às tarefas é integrado, principalmente pela utilização de bases de dados comuns e comunicação entre os elementos e o ambiente de gerência.

- O modelo é genérico em relação ao gerenciamento de redes diferentes. Foram propostas duas arquiteturas para gerenciamentos de redes ATM e IP, respectivamente. Ambas as arquiteturas são factíveis, porque utilizam elementos existentes tanto de uma rede como de outra.
- O modelo é adaptável e expansível, de forma que funcionalidades originalmente não previstas possam ser suportadas através da criação de novos módulos. Como os módulos existentes dão suporte apenas à gerência de QoS, em ambiente onde a gerência de segurança, por exemplo, é importante, o modelo pode ser expandido de acordo com as necessidades do administrador.
- A implantação de uma solução para gerência de QoS que siga o modelo proposto é mais complexa de ser executada que a implantação de uma solução de gerência padrão. Entretanto, essa complexidade é enfrentada pelo administrador apenas no início da gerência da rede. Durante a utilização do sistema, a complexidade é menor que a complexidade de uma solução padrão porque através de políticas o administrador exprime apenas o que deve ser feito pela rede, e não como deve ser feito (isso é de responsabilidade do sistema de gerência). Com a implementação de abstrações, o sistema pode tornar-se ainda mais amigável, agilizando as tarefas de gerenciamento de QoS.
- O modelo segue os padrões definidos dentro do IETF (como a utilização de protocolos padrão na implementação de sistemas de gerência), mas o suporte interno ao modelo nos equipamentos gerenciados, com a inclusão de elementos do modelo em cada dispositivo, é complexo, e tal complexidade é o resultado da indefinição do IETF em relação à padronização de questões críticas como a utilização de políticas e à monitoração da rede.
- O modelo é escalável em relação à vários aspectos. Inicialmente é escalável em relação a implantação de políticas na rede, pois as utilização de diversos consumidores de políticas distribuídos permite uma acomodação melhor do modelo, mesmo em rede grande. É escalável em relação à monitoração de QoS porque diversos monitores também podem ser utilizados em redes maiores, e por fim é escalável em relação à descoberta de QoS, pois a flexibilidade no número de identificadores de alvos utilizados permite ajustar a "velocidade" da descoberta de serviços.
- A escalabilidade do modelo é diretamente relacionada com a distribuição de seus elementos. Neste sentido, pode-se concluir que nem todos os elementos do modelo podem ser distribuídos. A base de dados de políticas pode ser distribuídas, assim como os elementos intermediários (consumidores de políticas, monitores de QoS e identificadores alvos), mas a base de *status* e principalmente a base de associações comportam-se melhor se forem centralizadas. Apesar disto, o modelo permanece escalável porque mesmo em redes maior as bases não são prejudicialmente afetadas pelas proporções do gerenciamento.
- O modelo é exeqüível, e isto é conclusão direta da implementação do ambiente de gerência QAME (*QoS-Aware Management Environment*), criada para gerência de QoS em redes IP.

Desde conjunto de conclusões outras importantes podem ser determinadas. Para que o modelo seja utilizado em redes diferentes, tanto em proporções quanto em tecnologia, o mesmo deve ser flexível, como verificado. A flexibilidade permite ao administrador a utilização de diversas configurações de utilização dos elementos do modelo de forma que sua rede seja mais bem gerenciada. A flexibilidade, por este aspecto, é uma característica positiva. Por outro lado, a grande liberdade de configuração pode tornar a implantação do modelo em uma rede uma tarefa complexa ao administrador, já que diversas escolhas sobre como os elementos do modelo devem ser dispostos na rede são deixadas para o administrador. Em certas situações, o administrador pode tomar decisões errôneas que podem ser implementadas porque o modelo é flexível, mas que neste caso não estaria operando de forma otimizada.

Em uma generalização, pode-se observar que o modelo implementa um gerenciamento por delegação, onde tarefas de gerência são delegadas a entidades distribuídas na rede que se comportam como gerentes intermediários. Os benefícios de um gerenciamento por delegação já são bem conhecidos pela comunidade de gerência de redes, e fortemente apoiados pelo IETF. Neste contexto, pode-se dizer que o modelo de gerência integrado proposto é um caso particular do gerenciamento por delegação, onde as tarefas de gerência de QoS suportadas são as seis definidas na classificação apresentada.

Por fim, através das contribuições desta tese pode-se concluir que a classificação das tarefas de gerência de QoS, o modelo de gerência integrada proposto, e a implementação do protótipo QAME para redes IP apresentada são indicadores de que a gerência de QoS, além de ser uma necessidade, é possível de ser realizada de forma integrada.

Outros trabalhos relacionados à gerência de QoS no modelo proposto ainda merecem mais investigações. Um estudo da representação dos dados de gerência deve ser realizado para que os elementos do modelo passem a utilizar um formato padrão de representação de informações. Algumas implementações iniciais indicam que XML pode ser eficientemente empregado nesta situação, onde dados de fontes diferentes (MIBs, bases de dados, diretórios) são representados uniformemente.

Como analisado, o nível de abstração fornecido no ambiente de gerência do modelo pode permitir uma gerência mais eficiente do QoS da rede. A investigação de mecanismos de abstração mais complexos deve ser realizada para que implementações do modelo (por exemplo, com novas versões do ambiente QAME) forneçam um número maior de facilidades ao administrador. O ponto crítico atualmente que precisa ser atacado em primeiro lugar é o fornecimento de mecanismos que permitam ao administrador a rápida determinação dos alvos em que uma política deve ser aplicada. Como visto, em redes pequenas isso não é uma questão crítica, mas em redes um pouco maiores esta determinação é normalmente problemática e pode complicar a gerência de QoS.

Anexo 1 Sistemas de apoio desenvolvidos

A tese apresentou os estudos realizados em relação ao gerenciamento de QoS, bem como propostas teóricas de organização de tal gerenciamento. O ambiente QAME visa suportar as seis tarefas de gerenciamento definidas. Este anexo apresenta as implementações realizadas como parte da tese de doutorado. Os protótipos implementados permitiram uma análise mais real dos conceitos apresentados.

NetPlus e gerenciamento padrão de redes

Como já discutido, o gerenciamento de QoS é um complemento ao gerenciamento padrão de redes. Isso significa que o gerenciamento padrão pode ser aplicado sem a existência do gerenciamento de QoS, ainda que isso seja ineficiente. Por outro lado, o gerenciamento de QoS só poderá existir em conjunto com o gerenciamento padrão. Logo, gerenciar uma rede com QoS envolve a utilização de gerenciamento padrão e de QoS, de forma complementar.

O sistema NetPlus [GRA 2001c] [GRA 2001d] permite a execução do gerenciamento padrão de uma rede, com vistas ao gerenciamento de QoS. As funcionalidades atualmente implementadas permitem o acesso aos dispositivos de rede através de Telnet, e SNMP. O gerenciamento de QoS é conseguido através da utilização de MIBs SNMP específicas.

O NetPlus é construído sobre o sistema operacional Linux [WEL 95] utilizando o servidor Web Apache [APA 2001]. Os pacotes de *software* complementares instalados no servidor do sistema são:

- PHP4 [PHP 2001] – Linguagem de *script* utilizada na implementação dos processamentos no servidor;
- NET-SNMP [UNI 2001] – Pacote de *software* que implementa o suporte ao protocolo SNMP integrado à linguagem PHP;
- MySQL [YAR 99] – Banco de dados utilizado para armazenar informações sobre a topologia da rede gerenciada, dispositivos e permissões de acesso dos usuários (gerentes);
- MRTG [OET 2001] – Pacote de *software* utilizado para monitoração das interfaces dos dispositivos.

O NetPlus é utilizado para o gerenciamento padrão da rede MetroPOA [GRA 99], cujo *backbone* é implementado com tecnologia ATM [DUT 95]. Os enlaces críticos são monitorados via MRTG e o acesso direto aos dispositivos é realizado através da navegação Web pela topologia da rede graficamente apresentada no sistema.

A FIGURA 1 apresenta a interface principal do NetPlus, após o sistema de autenticação ter validado o usuário e senha do gerente da rede. À esquerda do ambiente, encontra-se o menu do sistema que permite:

- Acesso à topologia da rede;
- Inclusão, remoção e alteração de topologias;
- Inclusão e remoção de dispositivos nas topologias existentes;
- Inclusão e remoção de enlaces entre dispositivos; e
- Inclusão, remoção e alteração de suporte às MIBs SNMP.

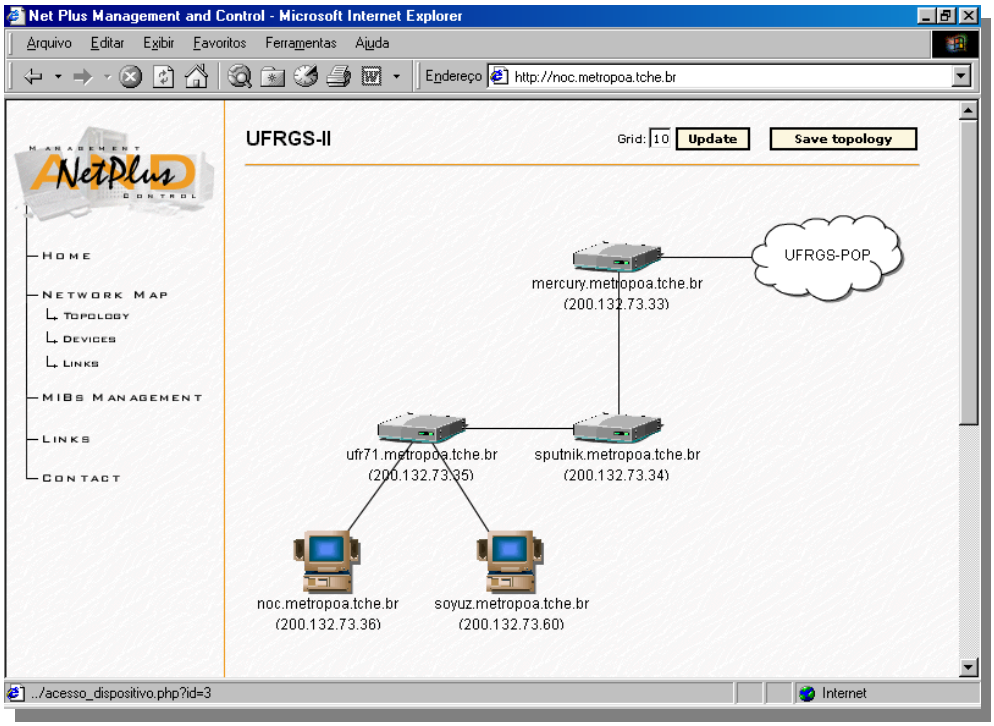

FIGURA 1 – Interface Web do sistema NetPlus

A apresentação gráfica das informações de gerenciamento nos navegadores requer a utilização das seguintes tecnologias:

- Flash [MAC 2001a], que permite a visualização de topologias, e alteração da disposição gráfica dos elementos (nuvens, dispositivos, etc.);
- JavaScript [WIN 2000], que complementa as interações implementadas em Flash e permite o desenvolvimento de interfaces de usuário mais sofisticadas.

A principal contribuição retirada da implementação do NetPlus é em relação à visualização das informações de gerenciamento. As soluções de gerenciamento baseadas na Web, tipicamente, apresentam a topologia da rede de forma estática, não permitindo a alteração interativa desta no navegador. Além disso, como as topologias são baseadas em arquivos de imagens, o volume de dados transferidos acaba sendo considerável.

No NetPlus a topologia da rede é montada diretamente no navegador. O servidor Web é responsável por transferir: a apresentação em Flash (atualmente em torno de 16 Kbytes) que desenha a topologia no navegador; e informações vetoriais sobre a localização dos dispositivos na topologia. Dessa forma, as informações transferidas do servidor Web para o cliente são em menor volume, se comparadas com as informações transferidas neste mesmo contexto por outras soluções de gerenciamento baseadas na Web. Por exemplo, uma imagem que apresenta a mesma topologia da FIGURA 1 criada no servidor tem tamanho aproximado de 50 Kbytes.

Network Executive e PBNM

O Network Executive [COE 2001] [COE 2001a] é uma ferramenta que implementa gerenciamento baseado em políticas (PBNM) em um nível de abstração simples, onde as políticas definidas são especificadas através da determinação de:

- Campos do cabeçalho IP para a definição de fluxos ou agregados;
- Parâmetros de vazão, atraso, *jitter* e perda para definição de classes de serviços;
- Associação entre os fluxos/agregados com classes de serviços.

Como cada política é traduzida nos consumidores de políticas, o sistema permite o registro dos consumidores que podem ser utilizados nas implantações. Da mesma forma o Nework Executive cadastra os alvos em que as políticas podem ser implantadas e as associações entre alvos, consumidores e políticas existentes.

Diferentemente do sistema NetPlus, o Network Executive é baseado em tecnologia Java para acesso via Web das funcionalidades de gerenciamento. Uma página Web de verificação autentica o gerente da rede, e uma console Java é aberta na máquina cliente. Conseqüentemente, exige-se que o navegador Web utilizado possua uma máquina virtual Java associada. A FIGURA 2 apresenta a interface gráfica da console do Network Executive.

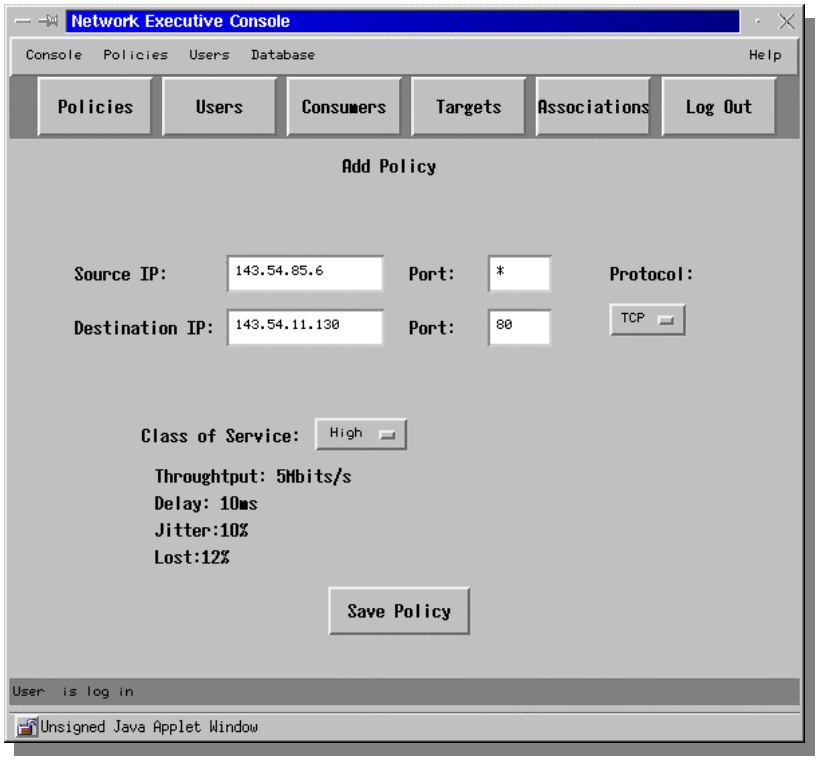

FIGURA 2 – Console de operação do Network Executive

O esquema de dados utilizado para o armazenamento das políticas é uma simplificação do modelo proposto pelo IETF [SNI 2000], pois o objetivo inicial da ferramenta era a execução de testes simples. Os dados informados pelo administrador de rede no console são transferidos ao servidor via protocolo próprio e armazenados em uma base LDAP. Para a recepção dos dados, junto ao servidor Web encontra-se um servidor específico (*Network Executive Daemon*) que aguarda novas políticas criadas. Tal servidor acessa então o servidor LDAP e armazena a política definida. O caminho inverso acontece quando o gerente da rede solicita a consulta de uma política. A FIGURA 3 resume a arquitetura interna implementada no Network Executive.

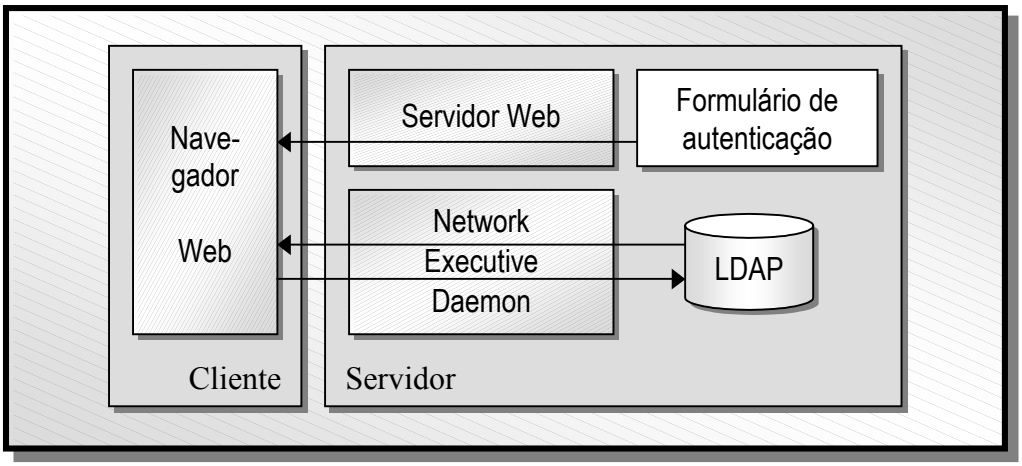

FIGURA 3 – Arquitetura interna do Network Executive

Os seguintes itens resumem as principais conclusões obtidas a partir da implementação do Network Executive:

- Ainda que o código seja otimizado, a tecnologia Java exige mais capacidade de processamento na máquina cliente se comparada com soluções não baseadas em Java, como o NetPlus;
- O uso de um servidor específico para transações de gerenciamento, como o implementado no Network Executive, libera o servidor Web para outras tarefas. Entretanto, nem todo cliente consegue acesso ao servidor específico principalmente quando da existência de filtros e *firewalls* no caminho entre cliente e servidor;
- A definição de políticas, ainda que em um nível de abstração não muito alto, é intuitiva e facilita a manutenção de operação da rede. Como a rede irá cumprir as políticas especificadas, é uma complexidade resolvida nos consumidores de políticas e nas arquiteturas de fornecimento de QoS utilizadas.

Monitor de QoS

O último protótipo implementado foi um monitor de QoS [RIB 2001] [RIB 2001a]. A implementação do protótipo visou analisar as dificuldades envolvidas na monitoração e verificar as formas de transferência entre as informações de monitoração definidas pelo gerente da rede.

Foram implementados um gerente e agentes monitores. A FIGURA 4 apresenta o ambiente onde as implementações foram utilizadas para teste.

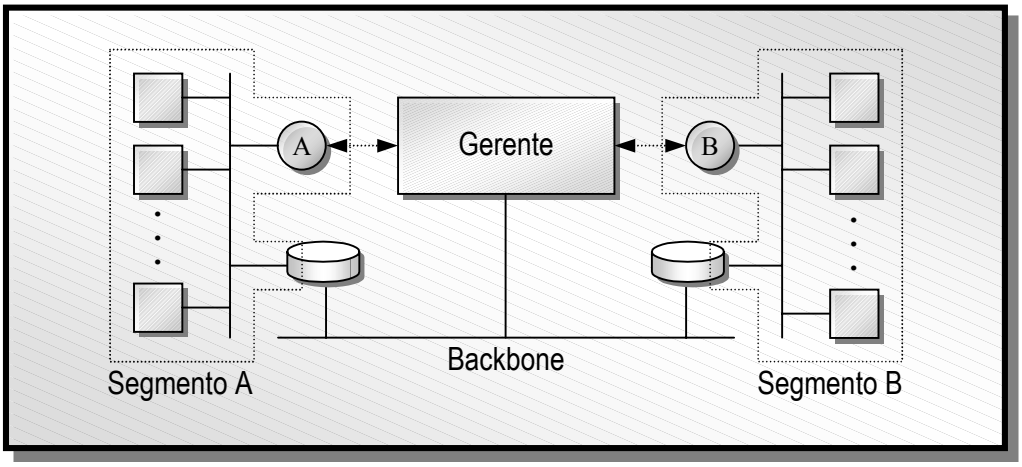

FIGURA 4 – Utilização do monitor de QoS

A rede utilizada é constituída de dois domínios de colisão, formando os segmentos A e B. Cada segmento possui diversas estações de trabalho representadas pelo quadrados da FIGURA 4. Em cada segmento foram colocados dois agentes monitores, representados pelos círculos. Cada agente monitor é, na verdade, uma estação de trabalho com interface de rede, operando em modo promíscuo.

Os segmentos foram conectados a um *backbone* principal através da utilização de dois roteadores com duas interfaces no mínimo. Além dos roteadores, o *backbone* possui uma estação que contém o gerente da rede, representado pelo retângulo central. As linhas tracejadas indicam as comunicações indiretas entre os agentes monitores e o gerente.

Nos testes realizados foram monitorados diversos fluxos/agregados IP. Para tal, a definição de cada monitoração foi realizada através da determinação de:

- Endereços IP origem e destino;
- Portas (TCP ou UDP) de origem e destino;
- Protocolos de transporte (TCP ou UDP);
- Valor do campo ToS (*type of service*) ou DS (*differenciated services*) na arquitetura de serviços diferenciados.

Da análise direta nos agentes monitores pôde-se verificar a taxa de ocupação de banda dos fluxos e a presença de *jitter*. A associação das informações coletadas nos agentes monitores permitiu, no elemento centralizador, a verificação do atraso e taxa de perdas dos fluxos/agregados monitorados. É importante ressaltar que na rede monitorada não havia sido instalada nenhuma arquitetura de fornecimento de QoS, como por exemplo serviços diferenciados ou serviços integrados. Logo, o monitor de QoS pode ser utilizado para verificação dos fluxos/agregados mesmo em uma rede onde não existe uma arquitetura de fornecimento de QoS.

As análises realizadas foram importantes, mas estavam limitadas ao tipo de implementação dos elementos monitores, que no caso se constituíam de estações de

trabalho com interfaces de rede em modo promíscuo. Isso limita a precisão dos resultados obtidos, visto que o tempo de verificação dos pacotes monitorados em cada agente é diferente do momento real em que o pacote deixou a estação origem, ou chegou a estação destino.

Uma monitoração mais efetiva pode ser conseguida se os agentes monitores fossem implementados diretamente nos roteadores. As MIBs de serviços diferenciados e integrados em processo de padronização no IETF fornecem informações importantes sobre os fluxos/agregados repassados e podem ser muito úteis, implementando pseudomonitores. Pequenas extensões a tais MIBs podem constituir agentes monitores completos.

Como uma rede monitorada nem sempre é baseada em *hubs* (caso da rede de teste utilizada) os agentes monitores implementados seriam de pouco valia em ambiente baseados em *switches*. Neste caso, a monitoração deveria ser executada também diretamente nos *switches*, e novamente as MIBs anteriores poderiam ser utilizadas como monitores mediante pequenas extensões anteriormente citadas.

Por fim, para se ter uma monitoração fim-a-fim precisa, agentes monitores também deveriam ser implementados diretamente nas estações origem e destino dos fluxos. Entretanto, esta tarefa seria mais difícil de ser cumprida, já que as pilhas de protocolos das estações deveriam ser alteradas em cada implementação de sistema operacional possível.

Implementações, tarefas de gerenciamento de QoS

As implementações apresentadas anteriormente procuraram verificar como partes do gerenciamento de QoS comportam-se em ambiente real. Esta seção apresenta o relacionamento entre as tarefas de gerenciamento de QoS definidas e as implementações realizadas.

A visualização de QoS foi principalmente verificado no sistema NetPlus, através da utilização das tecnologias PHP e Flash. Visualizações mais específicas devem ser definidas, mas as implementações no NetPlus indicam que uma adequada visualização pode ser conseguida com as tecnologias utilizadas.

A análise de QoS também foi verifica no NetPlus através da utilização da monitoração dos enlaces da rede MetroPOA. O tipo de gráfico criado por *softwares* como o MRTG e Cricket já é hoje adequado à análise de QoS. Informações mais precisas devem ser coletadas nos agentes monitores, mas a representação dessas informações já pode ser concluída como sendo apropriada.

O gerenciamento baseado em políticas foi investigado através do sistema Network Executive. A definição das políticas, atualmente empregada, é uma simplificação das propostas do IETF. Apesar disso, pôde-se verificar o comportamento real de bases LDAP e da efetiva aplicação de políticas na rede. Uma linguagem mais abstrata de definição de políticas deve ser definida, juntamente com um incremento na representação das mesmas. Existe também a necessidade de utilizar tecnologias mais adequadas na implementação do ambiente de interação com o usuário, já que as implementações em Java criadas se mostraram não adequadas, principalmente em ambientes onde existem filtros e *firewalls* entre o gerente da rede e a rede gerenciada.

A monitoração de QoS foi analisada na última implementação. Foi possível concluir que a monitoração através do esquema de agentes monitores é efetiva. Entretanto, a qualidade da monitoração é maior se os agentes monitores forem implementados dentro dos dispositivos de rede (roteadores e *switches*) e dispositivos finais (estações de trabalho). A integração da monitoração com a definição de políticas é uma necessidade atual. As políticas definidas devem ser traduzidas pelos monitores de QoS, de forma que o gerente não precise identificar os fluxos a serem monitorados através da definição de campos do protocolo de rede.

Anexo 2 Principais artigos publicados

Neste anexo são apresentados os três principais artigos publicados durante o doutorado. Os artigos refletem os estudos realizados em cada uma das etapas da determinação e investigação do problema, assim como a sua solução.

Managing Differentiated Services QoS in End Systems using SNMP

O artigo "Managing Differentiated Services QoS in End Systems using SNMP" é resultado dos primeiros dois anos de pesquisa em gerenciamento de QoS. É mostrada uma solução de gerenciamento de QoS em sistemas finais que utiliza o protocolo SNMP para programar como um transmissor deverá atribuir uma classe de serviços a cada pacote IP transmitido. É importante verificar que apenas o processo de marcação definido nos serviços diferenciados é suportado. Espera-se que a rede de computadores execute os outros processos envolvidos no fornecimento de QoS (por exemplo, conformação de tráfego, policiamento, prevenção de congestionamento).

Um mecanismo de solicitação de classes de serviços nos sistemas finais a um negociador de banda (BB) também é apresentado. Novamente o SNMP é utilizado como forma de comunicação, neste caso entre o solicitante (o sistema final) e o autorizador (o negociador de banda).

O artigo relata uma forma de gerenciamento de QoS ainda inicial, em um nível de abstração baixo, sem o uso, por exemplo, de políticas. Entretanto, as implementações apresentadas são utilizadas pelos elementos intermediários QAME (consumidores de políticas, monitores de QoS e identificadores de alvos).

Título:	Managing Differentiated Services QoS in End Systems using SNMP
Evento:	IPOM 2000 - IEEE Workshop on IP-oriented Operations & Management
URL:	http://ipom2000.kt.agh.edu.pl/
Datas:	De 4 a 6 de setembro de 2000
Local:	Cracóvia, Polônia

Integrated Management of QoS-enabled Networks using QAME

Também é apresentado o artigo intitulado "Integrated Management of QoS-enabled Networks using QAME" que apresenta o ambiente QAME descrito anteriormente na tese. O artigo é basicamente uma discussão sobre as seis tarefas de gerenciamento de QoS apresentadas do capítulo 3 e a apresentação da arquitetura e dos elementos QAME do capítulo 4. Por restrição de espaço, o artigo não discute as soluções atuais apresentadas no capítulo 2. Da mesma forma, os elementos

intermediários QAME não são mostrados de forma tão detalhada como a existente no capítulo 4.

Este artigo é resultado das pesquisas realizadas até o primeiro semestre do terceiro ano de doutorado. Como conseqüência, o artigo ainda não apresentava determinados resultados mostrados na tese, principalmente em relação às arquiteturas dos elementos QAME e em relação às implementações realizadas até o momento.

Título: Integrated Management of QoS-enabled Networks using QAME

Evento: ICN 2001 – International Conference on Networking

URL: http://iutsun1.colmar.uha.fr/ICN01.html

Datas: De 9 a 13 de julho de 2001

Local: Colmar, França

An Approach for Integrated Management of Networks with QoS using QAME

O último artigo do anexo, entitulado "An Approach for Integrated Management of Networks with QoS using QAME" [GRA 2001g], foi aprovado para publicação no DSOM 2001 (Workshop On Distributed Systems: Operations & Management).

O artigo é resultado do último ano de doutorado e apresenta o ambiente QAME com um conjunto de detalhes de implementação mais precisos. As tarefas de gerência de QoS, entretanto, não são discutidas, mas referenciadas. É também dado enfoque maior à solução de gerência baseada na Web aplicada no QAME.

Título: An Approach for Integrated Management of Networks with QoS using QAME

Evento: DSOM 2001 - IFIP/IEEE International Workshop On Distributed Systems: Operations & Management

URL: http://www.dsom2001.org/

Datas: De 15 a 17 de outubro de 2001

Local: Nancy, França

MANAGING DIFFERENTIATED SERVICES QOS IN END SYSTEMS USING SNMP

Lisandro Zambenedetti Granville, Rodrigo Uzun Fleishmann
Liane Margarida Rockenbach Tarouco, Maria Janilce Bosquiroli Almeida
Institute of Informatics, Federal University of Rio Grande do Sul
Av. Bento Golçaves, 9500 – Porto Alegre – Brazil
e-mail: {granville, uzun, liane, janilce}@inf.ufrgs.br

ABSTRACT

Offering QoS on TCP/IP networks is the object of intense research. New applications, such as telemedicine, distance learning, videoconference and others can only be implemented on environments that ensure QoS for the existing services. Usually, routers are the network resources that undergo changes in order to implement it – hosts have little or no influence. This paper presents an implementation of QoS structures offered directly from the host. It also presents a marking process that is run on the final systems and can be remotely managed via SNMP. A signaling application is also described in order to allow QoS requests to inherited applications that do not perceive QoS service on the networks.

1 INTRODUCTION

The best-effort paradigm currently found on TCP/IP networks is able to offer enough services to support the great number of applications spread on the Internet. However, an important set of applications can not operate properly on best-effort services only. Applications such as videoconference, distance learning and telemedicine, for instance, will only operate on environments where services are guaranteed. Given this setting, one of the great current challenges is to offer QoS [1] on TCP/IP networks.

One of the apparently promising propositions is the use of differentiated services architecture [2] of IETF [3]. Its importance is clear since the Internet2 project [4] has elected differentiated services as the solution for QoS supply. In addition, several manufacturers have supported the solution, and software houses are making support tools available on initial versions (for example, Bandwidth Brokers).

The main solution to offer QoS is through network equipment. Currently, hosts perform little or no role in it, leaving the routers with most of the work to be done. On the other hand, chances of success increase with the distribution of QoS tasks. Therefore, it is reasonable to think that involving hosts is also important to run the services properly.

This paper presents an implementation of QoS structures based on host implementation. The packet marking process is now run directly on the final systems, separately from the applications. The network manager can program all hosts remotely and control the marking process globally, exchanging SNMP (Simple Network Management Protocol) messages [5] with the hosts. A Marking MIB (Management Information Base) [6] has been implemented so that the manager can program the hosts.

QoS management on the hosts can also be automated to the extent of replacing the management station with a Bandwidth Broker (BB) [7]. Users can request QoS parameters to the BB. This request uses a signaling tool. Resources can be obtained immediately or scheduled for future use, according to the capability of the BB. The request communication also uses SNMP and a QoS MIB.

This work is organized as follows. Section 2 presents the packet marking process on the hosts. Such remote marking management uses the Marking

MIB and the SNMP marking support shown in section 3. Section 4 presents considerations on user resource requests. Section 5 describes the QoS MIB that implements the host/BB signaling. Section 6 presents management and QoS supply scenarios that may benefit from this work. Finally, section 7 wraps it up and presents guidelines for future studies.

2 PACKET MARKING ON END SYSTEMS

The process of marking IP packets through the DS Field allows for the differentiation of data flow on leaf routers and differentiation of aggregate flow on other routers. Marking is based on service classes that represent a set of previously defined QoS parameters, which does not mean such classes are static.

The packet marking process is performed normally on the first routers close to the flow sources (leaf routers). On the other hand, aiming at providing data flow QoS throughout the whole transmission path (from data transmitting host to receiving host), this work suggests that the marking process be performed on the hosts. As a consequence, leaf router activity is minimized, which frees the routers to process other functions.

Also, since this host marking process offers marked packets from their source, the first flow segment can make use of level three switches that run traffic conformation and prioritization functions. This enables QoS as early as on the local network, which can be relevant for a context of applications that require QoS but do not have flow beyond the first router.

2.1 Applications Should not Mark

Who in the host is to mark the packets? Some solutions suggest that applications assign a service class to the flows they generate. This could be done by marking the IP packets that the application directly generates. We understand this should not happen for two main reasons: architecture and management.

The DS Field belongs to the IP packet (network level). As such, only the IP layer of the protocol hierarchy should access it. If the application could access this field, hierarchy would be broken, since the transport level (between the application and network levels) would not be used. Besides that, as a rule, applications do not have direct access to IP packets. Rather, applications use API interfaces so that programmers are free from network implementation details.

However, the worst problem with application packet marking is related to management. Let's examine what could happen during the development and testing of network applications. A programmer develops an application with no concern about the priority of application flows and notes that the performance is not satisfactory. In an attempt to optimize communication and having the ability to mark, the application will start generating high priority packets and performance will increase. All developers want their applications to have optimal network data flow performance. If all applications generate high priority flows, the result will be similar to best-effort because several flows will fight for the same high priority treatment. This is certainly not advisable. Packet marking control should not be left to applications, but to a neutral, resource management entity capable to determine if the network can deal with a new flow. Therefore, the marking process must be implemented on a different context than the application.

2.2 Independent Applications

Nowadays, there are several applications that use network resources: plant production line monitors – called supervisors –, financial transaction applications, e-mail clients, web browsers and so on. Some of these use more and others use fewer network resources. This work does not intend to ignore these existing applications. On the contrary, one of the key points was not to cause changes to the applications because of the model, or at least minimize the adaptation effort to fit to the proposed architecture. Several architectures have been considered so that applications run regardless of QoS on the network. A first distinction can be made between two large groups: networks that implement QoS and networks that do not.

The first group can be further divided into hosts that implement QoS and hosts that do not. The former can be subdivided again into applications that perceive QoS and applications that do not. In every case, it is desirable that the application does not have to change its source code. In all situations, we will show that the source codes will not need changing.

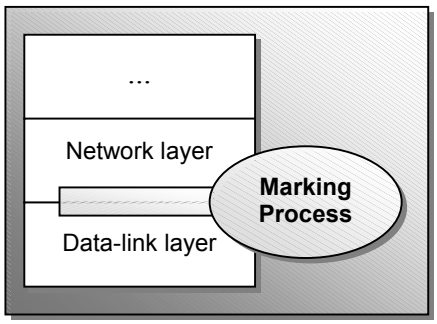

Figure 1. Protocol Stack and Marking Process Architecture.

2.3 The Marking Process and the Protocol Stack

Since the marking process that differentiates data flows will not be implemented by the application, we believe the best way to do so is to implement the marking process on the host protocol stack. It is important to notice that the packet is only marked after the network layer has generated the IP packet and before the data-link layer has received the data (Fig. 1). The implementation adds a module that captures the packet sent by the network layer to the data-link layer in a way that neither perceives the marking process.

3 MARKING MANAGEMENT ON END SYSTEMS

The packet marking process on the hosts is performed by looking up a local database that indicates how the packets generated by the host should receive the DS Field values. Programming this base allows changing packet marks and new flows can be identified and receive appropriate DS Field values. Packets that do not have a matching rule on the database have a programmable default handling, usually best-effort (setting zero to the DS Field).

The database can be programmed on the specific corresponding host. However, that programming requires the network manager's presence, which can be impossible where environments have a lot of equipment. Thus, we propose a solution for programming and managing the host marking process by accessing a Marking MIB (Fig. 2).

3.1 Objectives

On a network management environment, the manager must access resources remotely. This access allows for listing the configuration of equipment and detecting faults. According to the information obtained from the resources, the network utilization ought to be optimized by performance management. The resources used can be equipment (routers, switches, hosts) as well as services (web servers, databases, news servers). Considering the packet marking process on a host as a service available on that host, one must implement mechanisms to manage this service.

Once these mechanisms are present, a sequence of management forms can be implemented. Initially, central management can statistically determine the programming of every host with a marking process. Where hosts have inadequate flow marking, the manager can remotely reconfigure hosts through a management platform. In more dynamic environments, a Bandwidth Broker (BB) substitutes for the management platform [7]. Host programming becomes automatic through the use of management policies on the BBs.

Finally, remote access to host programming should be such as to integrate management and other network features. The network manager should be able to control and monitor equipment, services and the marking process by using the same management environment. In order to do so, there must be a standard method to access hosts. Our solution uses the SNMP protocol to communicate manager and host marking process, and a Marking MIB to define what information the manager can access and program.

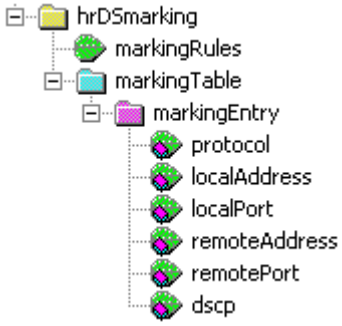

Figure 2. Host Marking MIB

3.2 The Marking Process and the SNMP Protocol

In order to program the marking process remotely, it is necessary to support the management protocol chosen – SNMP. This can be achieved through direct or indirect integration.

Direct integration involves the programming of the marking process to install SNMP support. The marking process observes its own internal configuration and returns the requested information to the requesting manager. Where there is an attribution, the marking process starts marking packets according to the performed programming and updates the database so that programming is consistent (i.e., not lost next time the host is restarted).

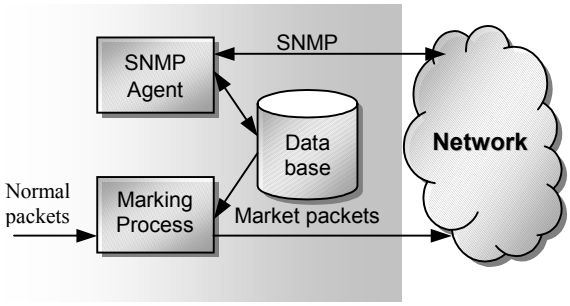

Figure 3. Indirect integration between SNMP agent and marking process.

Indirect integration does not need any changes in the marking process. However, there must be a new process to support SNMP exclusively. Integration is indirect because the marking process communicates with the SNMP support through the database. Upon request, an SNMP agent returns the values found on the base. When it is programmed, the SNMP agent changes the existing data values (Fig. 3). The marking process perceives these changes every time a new packet is marked on the host. Because the marking directly sees the base alteration, there is little or no performance loss in indirect integration. Our solution uses indirect marking so that packet marking and database programming via SNMP remain independent.

4 USER-BASED SIGNALING

QoS requires a signaling process that is used to reserve network resources. This reservation ensures that the requested QoS parameters can be effectively supplied. Signaling can be done in different ways, depending on the QoS architecture under use. For instance, ATM network signaling requires resources to be reserved in each node between source and destination of a flow. In the differentiated services architecture, signaling is performed by the BBs of the domains located between source and destination.

Signaling must start at the source of the flow and continue to the destination reserving resources. At the source, the application that perceive QoS services on the network have to use programming APIs to start the request. One of the problems is how to let applications that do not perceive QoS benefit from the network services without having their code changed.

4.1 Signaling and BBs

The differentiated services architecture includes a resource negotiation through communication with the BBs. Request messages are sent to a domain's BB, which verifies the availability of the requested resources. The received requests can originate in neighboring domain BBs or hosts of the domain itself. Therefore, there must be a signaling protocol to enable communication between neighboring BBs, and between hosts and a BB within the same domain (Fig. 4).

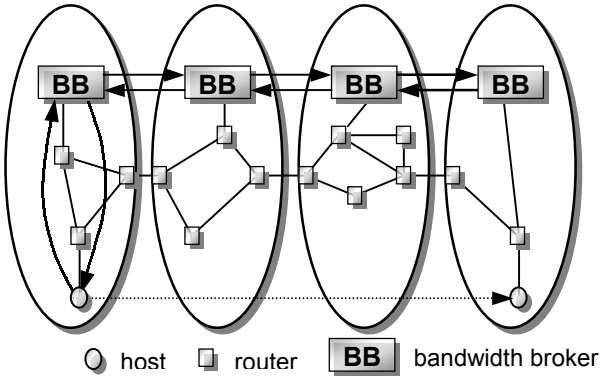

Figure 4. Signaling between hosts and BB, and between neighboring BBs.

Proprietary protocols can be built on TCP or UDP and used for this task. However, they may not be interesting because the communication between neighbouring BBs supplied by different manufacturers is limited since BBs are not likely to interact. The use of open standards increases the chances of interactivity between BBs. IETF [3] has workgroups that conduct research on such protocols. From a technical point of view, RSVP [8] is the most promising because it was developed aiming at reserving resources on a QoS environment. Another possible option is to use SNMP [5] as a protocol for requesting resources.

4.2 Are the BBs Management Systems?

The role of a BB is to receive requests, verify the resources of the local domain and make new requests towards the destination of a flow. If all resources requested to a destination can be allocated, each BB shall proceed network programming on its domain. This programming involves accessing information on the routers and configuring them so that they correctly handle the new data flow to be generated from the source.

Accessing and configuring routers are typical functions run by network management systems on best-effort environments. In this sense, one can say a BB is a network manager with a specific task: QoS management. The main difference is that whereas the management system typically needs human intervention to program the routers, a BB will do so automatically. Thus, we can say that, given this context, a BB is a management system specialized in configuring routers automatically.

A common issue to BBs and management systems is interactivity. Two management systems on different domains currently cannot communicate except on proprietary solutions. IETF faced this

problem when it started SNMPv2 work [9]. The definitions of new protocol functionalities planned for manager communication using InformRequest messages. The problem was not solved because the final protocol adopted by the industry was SNMPv2c [10], which does not have such messages.

The interactivity between different domain BBs is similar. Since there is not a standard protocol to exchange information between the BBs, proprietary solutions are used in spite of their incompatibility. Although it is a serious problem for both environments (management systems and BBs), it is more urgent to solve communication between BBs. Unless there is communication, it is not possible to ensure QoS between source and target located on different domains. We believe that solving the problem of communication between managers will eventually solve the problem of communication between BBs, which in its turn will define the communication standard between hosts and the BB of the associated domain. For the time being, our work present a communication between host and BB based on SNMP, as shown below.

4.3 Motivation

Our solution needs to look up on two main problems: offering a communication mechanism between host and BB and ensuring that applications that do not perceive QoS on the network services can use them anyway.

One way to offer QoS to applications that do not perceive this possibility is by having the manager identify flows statically. Imagine a critical database application such that can be programmed for the flow generated by a replication between 6 and 7 p.m. to have priority over the HTTP flow. On one hand, the network manager is not always able to determine which flows generated by a host are the most important. In this situation, it is the user who knows it and should decide the conditions of each flow. Thus, the most appropriate way to determine QoS parameters for each flow is to obtain this information from the user.

Item 2 presented a method to identify flows from their source, by marking packets on the hosts. It is also necessary to supply a mechanism to request resources on the hosts. This mechanism must be such that applications do not need changing. Supplying such mechanisms focuses on the host a considerable part of the differentiated services logic.

5 INTRADOMAIN SIGNALING PROTOCOL

Our solution uses SNMP protocol to request QoS parameters from the host to the BB. We believe SNMP can also be used for neighbouring domain BBs, but this is not within the scope of this study.

Host requests parameters by accessing a QoS MIB (Fig. 5) on the BB. Parameters are sent as a sequence of objects. When the ASK object (an integer) of the QoS MIB receives value 1, the request analysis process is started on the BB. When ASK is altered to 2, the analysis process has been completed successfully, which means the request has been accepted and the network programmed to support the new flow described. If ASK is 0, the BB has denied the request and the host can look up what parameters could be accepted by observing the new status of other objects of the QoS MIB.

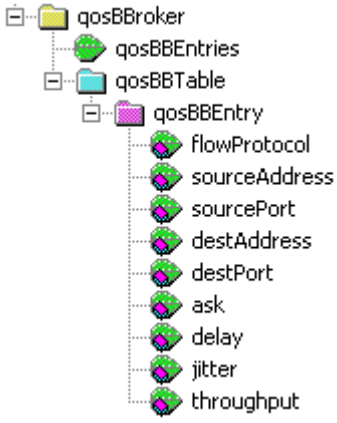

Figure 5. QoS MIB

5.1 Users Ask – Applications Get

It is the host that must access the QoS MIB described above. Applications that perceive QoS must proceed with SNMP request and monitor the MIB objects on the BB to check for the request return. If the request is accepted, the flows can rely on the network services since they will be programmed to support the flow described on the request.

A signaling tool has been developed for applications that do not perceive QoS (most of them). The host user uses the application to describe QoS parameters for applications that do not perceive such services on the network. Once parameters are described for each application, the user can proceed and request services to the BB. If the application started by the user is accepted, the network is programmed and the application can use the services. In short, the user defines which applications are important and which QoS parameters are most appropriate. Finally, the user sends the requests to the BB and the applications use the programming run on the network.

5.2 Implementation and Operations

The request tool is built as a standard application that uses SNMP as an information exchange protocol. Internally, the tool knows the QoS MIB on the BB. The QoS parameter requesting process takes the following steps:

1. **Selecting an application profile**. Some application profiles are previously supplied for more usual applications. For instance, a profile for POP3 clients describes an application that may have high delay and irrelevant jitter. Therefore, the discard priority can be high, since a late e-mail message does not threaten this application. On the other hand, the profile for a videoconferencing application requires low delay, controlled jitter, high flow and low discard priority.

2. If none of the existing profiles satisfactorily describes the requisites of an application, **the user can define new profiles** with appropriate parameters. The parameters of existing profiles can also be changed. It is important to notice that it is the user who decides on the appropriate QoS requisites. Pre-recorded profiles are only a starting point for a more precise definition.

3. Once the most appropriate profile for an application is determined, **the user must schedule the utilization time of the profile**. If immediate use is wanted, only the interval is to be set. As an option, the user can also assign the selected profile a process number to be monitored. If the process is ended before the profile interval finishes, it may be disabled as soon as this situation is detected.

4. **Request the QoS parameters stored on the profile to the BB**. This step is only taken on the application if the user has not scheduled a request and wishes to do it immediately. The tool will communicate with the BB and map user-supplied information onto objects of the QoS MIB. The tool will monitor the BB while waiting for a reply to the request.

When a request is scheduled, it will be run by the tool at the programmed time. Where requests are scheduled but not accepted by the BB, there is an event log that directs to the failure on the reservation process. In fact, scheduling can be carried out in two different ways, depending on the capacity of the BB:

a) **Immediate confirmation scheduling**. When a user requests scheduling, the tool checks the BB to find out whether the necessary resources will be available at the requested time. Upon affirmative response from the BB, the scheduled request will certainly be successfully run. In this case, the BB must be able to store information about reserved resources internally in order to determine whether resources will be available at a given moment. BBs that do not implement this solution are in the category presented below.

b) **Delayed confirmation scheduling**. If the BB can not immediately determine whether a scheduling can be run, the tool will request resources to the BB at the scheduled time. Denied requests will be informed through entries on a log file that the user can study.

6 PUTTING ALL TOGETHER

The local domain must have a BB that informs about the requested resources so that the signaling tool is used effectively. The BB implements the QoS MIB, which lets the signaling tool request and monitor resource requests.

Currently, no BB works according to our definitions. Therefore, there are two options to offer QoS on this environment: creating a proxy for the existing BB or creating a small testing BB. We have chosen the latter because it could be more promptly implemented in addition to providing our team with the experience of programming a BB. Our BB is simple because it implements the necessary QoS MIB, but it does not communicate with neighboring domain BBs. This communication is part of future developments and projects.

This BB has a resource scheduling feature that can be disabled. Scheduling allows hosts to mark resources that will only be used later. We have decided to allow disabling the functionality so that the host-based scheduling could be tested. In this case, the BB does not store any previous information about the resources, but the host waits and makes the request at the time set by the user (section 5.2).

Figure 6 presents a general diagram of our solution. The structures developed are enough to offer QoS from the hosts on different environments, as follows.

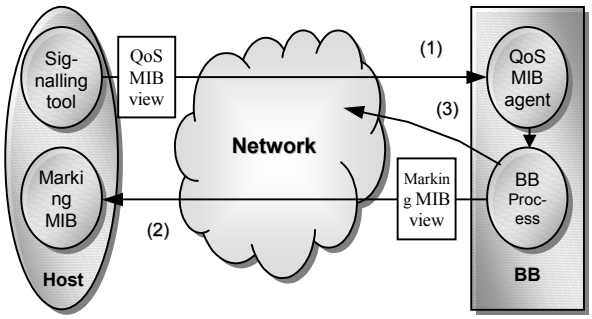

Figure 6. Management architecture of host-based QoS

6.1 Static Flow Priority and Human Network Manager

On a network environment where there is only one management station, the human manager is responsible for accessing the hosts that will generate the many network flows. Since there is not a BB, signaling between host and BB is not possible. The users know their needs and contact the network manager personally to inform their priority flows.

Upon receipt of user requests, the network manager determines the priority flows, where they come from and their destination. The most critical hosts (web servers, SMTP and news servers) will certainly be more important. Strategic users (company president, financial manager, and so on) also have priority over common users. Finally, critical applications (data base updating, e-commerce and videoconferencing) are chosen.

Once flows are qualified, priorities are attributed. Then, the manager starts programming hosts by accessing the Marking MIB (Fig. 6, number 2). The computer network is also statically programmed to meet the specifications of the qualified flows (Fig. 6, number 3)

6.2 Dynamic Flow Priority with Network Manager Scheduling

If the managing station has an application to automate network host programming, it is possible to optimize the environment described above. The manager can delimit critical periods when special flows should have differentiated priority. For example, the managing station tool can program the host that keeps the database so that it has maximum priority when base backup is run.

6.3 Dynamic Flow Priority with Signaling Scheduling

The most complete environment is that in which the managing station is replaced by a BB. Users request QoS parameters for their applications directly to the BB (Fig. 6, number 1). The BB checks whether the requested resources will be available at the set time. If this is not possible the signaling tool informs the user of the network capabilities available on the BB. Then, the user can proceed with a new request.

When possible, the BB will program the host (Fig. 6, number 2) at the scheduled time so that the requested flows take the appropriate QoS parameters. Network programming is also carried out (Fig. 6, number 3) so that the new flows can be effectively supported. On this last environment, QoS negotiation is dynamic and does not require intervention of a human manager, who keeps the usual management tasks.

The three environments described here show the possibilities of using the solution in different situations. There can be several other configurations depending on the necessity of each environment. The architecture of the solution allows the generation of differentiated traffic to be run from its source through the Marking MIB. The other structures (reduced BB and signaling tool) attempt to make the configuration of the host marking process easy and automatic.

7 CONCLUSIONS AND FUTURE WORKS

QoS management on hosts that are placed on a differentiated services network allows applications to operate more appropriately according to each user's specification. To do so, we have presented a host management architecture based on SNMP protocol and divided into two main elements: a flow marker and a signaling tool.

The flow marker lets the network manager program the marking process on the host, which involves attributing a value to the DS Field. That requires exchanging messages between the source and target IP protocol addresses, the transport protocol used (usually TCP or UDP) and the source and target transport protocol ports. Usually, the marking process is run by a router on the entry interface. Our solution improves the performance of leaf routers because packets are marked at their source (i.e., the hosts), which frees routers from this task. Because the marking process responds to SNMP messages, marking can be remotely controlled by a network management system (NMS) that accesses the Marking MIB on the host.

The second element is a signaling tool. The network manager can program the machines to prioritize certain flows, but it is the user who knows the needs of the applications on the hosts. In order to supply the manager with more precise information about how to deal with the several flows, the signaling tool is used. Users identify an application profile on their hosts and schedule a resource reservation. Also by using an SNMP protocol, the signaling tool requests the manager a special set of resources. Such request is obtained by accessing the QoS MIB that has been developed.

The management system that is used can automate the host marking process based on the requests of the signaling tool. This makes a human network manager unnecessary. Such automation can also be obtained by using Bandwidth Brokers (BBs), which receive requests from the signaling tools and negotiate resources with the neighboring BBs towards the destination of a flow. If the requested resources can be allocated, the BB also programs the source host to

mark the packets generated.

Future projects to be developed to manage QoS-perceiving hosts are currently related to two issues: security and applications. Security attempts to ensure that requests from a signaling tool are not monitored by sniffers. Also, it is important to certify the user who accesses the QoS MIB in order to ensure that only valid users can access information on the BB. The plan is to use the SNMPv3 protocol [11] to solve such matters, since it can encrypt messages and because its verification system is more sophisticated than the simple use of community strings found in SNMPv1 and v2.

Application-related matters aim at supplying communication mechanisms between new applications and the signaling tool. This will allow applications to request QoS parameters, but will not need the user to know the corresponding new parameters. Communication between new applications and the signaling tool can be done in several ways and encapsulated in API programming functions still to be developed. This will require the creation of communication libraries and the alteration of the signaling tool to allow external communication.

REFERENCES

[1] A. Campbell, G. Coulson, F. Garcia, D. Hutchison, H. Leopold, "Integrated Quality of Service for Multimedia Communications", *Proc. IEEE INFOCOM'93, pp. 732-739, San Francisco, USA, 1993.*

[2] DiffServ Workgroup Homepage. URL: http://www.ietf.org/html.charters/diffserv-charter. Html, 2000.

[3] IETF Homepage. URL: http://www.ietf.org, 2000

[4] Internet2 Homepage. URL: http://www. internet2.org, 2000.

[5] J. Case, M. Fedor, M. Schoffstall, J. Davin, "A Simple Network Management Protocol (SNMP)", RFC1157, 1992.

[6] D. Perkins, E. McGinnis, "Understanding SNMP MIBS", Prentice Hall, 1996.

[7] K. Nichols, V. Jacobson, L. Zhang, "A Two-Bit Differential Services Architecture for the Internet", RFC2638, 1999.

[8] R. Braden, D. Clark, S. Shenker, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification", RFC 2205, 1997.

[9] W. Stallings, "SNMP, SNMPv2 and CMIP", Addison Wesley, 1993.

[10] M. A. Miller, "Managing Internetworks with SNMP", 2nd edition. M&T Book, 1995.

[11] W. Stallings, "SNMPv3: A security Enhancement for SNMP", *IEEE Communication Surveys, 1(1), 1998.*

Integrated Management of QoS-enabled Networks using QAME

Lisandro Zambenedetti Granville, Liane Margarida Rockenbach Tarouco, Márcio Bartz Ceccon, Maria Janilce Bosquiroli Almeida

Federal University of Rio Grande do Sul – UFRGS – Institutes of Informatics
Av. Bento Gonçalves, 9500 – Block IV – Porto Alegre, RS – Brazil
{granville, liane, ceccon, janilce}@inf.ufrgs.br

Abstract. Providing QoS-guaranteed services in current installed networks is an important issue, but only the deploying QoS services is not enough to guarantee their success: QoS management must also be provided. Nowadays, policy-based management addresses this need, but such management is not enough either. Network managers often deal with QoS tasks that cannot be performed using only policy-based management. This paper describes six important QoS management-related tasks (QoS implementation, operation maintenance, discovery, monitoring, analysis and visualization) and shows solutions that can help managers proceed with these tasks. Unfortunately, these solutions are independent from each other, leading to a scenario where integration is difficult. To solve this lack of integration, QAME (QoS-Aware Management Environment) has been developed. QAME provides support to allow the execution of the defined QoS tasks in an integrated fashion.
Keywords: QoS management, policy-based management, management environment

1 Introduction

The great majority of today's networks operate based on a best-effort approach. There is no warranty concerning traffic delay, jitter and throughput. Several applications operate properly on this environment, but several others can only be delivered if network QoS (Quality of Service) warranties are present.

Managers should be aware of QoS features implemented on the network. QoS architectures can only be effective and provide guaranteed services if QoS elements are adequately configured and monitored. Thus, in addition to the management of traditional elements (router, switches, services, etc.), managers must also manage QoS aspects. In this scenario, it's not a surprise to realize that the overall management will be more complex.

Effective QoS management can only be achieved if QoS management tasks are properly identified. Based on these tasks, management mechanisms can be defined to help managers to deal with QoS elements. Besides, such mechanisms must allow the replacement of the current device-oriented management approach by a network-oriented approach. This approach replacement is necessary because the amount of management information is even greater when QoS-related data are present. Device-specific management takes too long and does not provide a global view, which is specially needed when monitoring QoS parameters in critical flows.

QoS management must also take place within the same environment used to manage standard network elements. Management platforms should be aware of QoS to help managers proceed with QoS related tasks. Unfortunately, today's management platforms are not QoS-aware and managers have to investigate each QoS-enabled device to check QoS parameters. Again, a network-oriented approach is needed, with features that explicitly

present QoS information to managers in an intuitive fashion.

This work presents our ongoing QoS management project named QAME (QoS-Aware Management Environment). The paper shows current analysis on QoS tasks and related software developed. The final goal is to provide an environment where managers are able to deal with explicit QoS information in a higher abstraction level, and with a global network view, implementing the network-oriented approach mentioned before.

The paper is divided as follows. Session 2 presents QoS-related tasks that should be performed to allow the deployment and maintenance of QoS architectures. Session 3 introduces our proposed environment. Finally, session 4 concludes the paper and also shows future works to be developed.

2 QoS management-related tasks

QoS-enabled networks can only present reliable QoS services if managers are aware of QoS. Each network can use one or more QoS solutions and protocols. For instance, one could use MPLS [1] within an administrative domain, and have agreements with neighboring domains using DiffServ [2]. Another could use RSVP [3] to dynamically reserve network resources for a scheduled videoconference. Every QoS-related element (routers, brokers, hosts, classifiers, markers, etc.) must be managed to guarantee proper QoS service operation.
In order to manage QoS elements, managers must perform several tasks, besides the tasks applied to traditional management. QoS-related tasks must be performed by using facilities provided by the network management environment, but today's platforms do not provide any explicit QoS facility. To start providing such facilities, firstly we need to identify QoS tasks and then check how facilities should be created to help QoS management. Thus, we have classified QoS management tasks as described in the following sub-sessions.

2.1 QoS installation

We call "QoS installation" the task of choosing QoS solutions and installing such solutions into a network. The result of QoS installation is a QoS-unique architecture that provides QoS services to a particular network. The resulted QoS architecture is a collection of QoS solutions and protocols that, together, offer services to network users.

Nowadays, the main QoS solutions offered are those developed under IETF: MPLS, DiffServ and IntServ [4]. Actually, each solution can be deployed by using different configurations. For instance, one could use bandwidth brokers [5] to allow dynamic DiffServ resource reservation. Another could use DiffServ with no brokers installed on the network.

There is a need for software that advises managers about possible QoS solutions, configurations and problems. One solution can be appropriate for a particular network, but totally inappropriate for another network. The final architecture must result from the analysis of the following elements:

- **Network topology**. Each network has a particular topology and for each topology a particular architecture is more suitable;
- **Network traffic**. Even with identical topologies, each network faces different traffic. One could have more downstream than upstream traffic, or more internal than external traffic;
- **Network application priorities**. Network traffic is mainly generated by applications (network itself doesn't generate too much traffic). The level of desired application priorities vary from network to network;

- **Network available solutions**. The final architecture can only be applied if selected solutions are supported by network devices. Otherwise, the final architecture has to be computed again.

Deriving the final architecture could be helped by QoS simulation solutions. Managers would describe its network topology, traffic and application priorities, and start checking each available solution. Different solutions could exist in the same environment at the same time, to collaborate with each other. For instance, a manager could use RSVP reservation internally and DiffServ on the network boundaries.

After deciding on a final architecture to be used, managers should start configuring devices, changing inappropriate ones, updating software on others, etc. Such procedures can be done in two different ways: locally or remotely. Local procedures are time consuming. Thus, we must provide tools able to keep local procedures minimal. Configuring and updating software can be remotely done on almost all cases. Except for IP number configuration and a few other operations, configuration is remotely driven, mainly through Telnet command line interface, HTTP in modern devices, and by using SNMP [6] agent/manager interaction. Changing equipment is an intrinsically local operation, and cannot be done remotely.

2.2 QoS operation maintenance

After QoS architecture is defined and installed, QoS services are ready to be offered to network users. At the same time QoS architecture is serving user needs, the manager must define appropriate operational parameters. We qualify any procedure taken to define QoS service behavior "on the fly" as "QoS operation maintenance".

Procedures often taken in QoS operation maintenance are those related to traffic classification, marking and prioritizing. Bandwidth static reservation, SLAs management [7] and policing are also examples of operation maintenance.

Today, the promising solution in QoS operation maintenance is policy-based QoS management. By using policies, managers can determine, at a higher abstraction level, how the QoS architecture should proceed to meet a desired behavior. Policy-based management actually constitutes a whole architecture itself, with dedicated elements (PEP and PDP [8], for instance). One important procedure in policy-based management is, for example, the location of policy enforcement points. Tools should also help managers in defining such points.

2.3 QoS discovery

Some network equipment is overfilled with features. It is not rare to see complex routers with several services being used only for packet forwarding. Although several features could be used to help QoS provisioning, they are not, since managers cannot handle so many device features in large, complex networks.

We define "QoS discovery" as the task of searching the network for features that can help or improve QoS provisioning. Normally, QoS discovery is done through SNMP messages sent to devices, or by using any other proprietary method. QoS discovery can also be performed checking equipment documentation, but that is not an operation that could be automated.

QoS discovery is helpful in two important moments: in QoS installation and in QoS operation maintenance. In QoS installation the new discovered features can be used to decide

among QoS architectures. In QoS operation maintenance, it can report new added equipment with QoS features while the QoS architecture is running.

QoS discovery can be effective if discovering mechanisms are installed. Such mechanisms can be simple (polling equipment trying to find MIBs related to QoS architectures) or complex (distributed network monitoring to check traces of critical protocols, such as IGMP [9] and RSVP [3]).

2.4 QoS monitoring

Managers have to be up-to-date about the difference between the desired QoS and the faced QoS. This difference can never be greater than a defined critical value. If the faced QoS is too different from the desired QoS, user applications would degrade indicating that the contracted services are not running properly.

To check the current QoS, managers should collect data in the network through QoS monitoring. This task has to be able to determine two related pieces of information:

- **End-to-end achieved QoS**. The QoS parameters of particular segments are important, but end-to-end QoS is crucial. If end-to-end achieved QoS is not correct, managers must be warned to fix the problem. End-to-end QoS degradation is the sum of the degradation of each segment on the end-to-end path. If just one segment is introducing degradation, that will be noticed in the end-to-end QoS.
- **Points of degradation**. If end-to-end QoS degradation is noticed, QoS monitoring should be able to determine where in a flow path the degradation points are.

Today, most QoS monitoring solutions are only able to satisfy the first item. It is simple to check if there is end-to-end degradation, by using RTP/RTCP [10] protocols, for example. However, identifying degradation points is a more complex procedure, and requires more complex processing on the network [11].

2.5 QoS analysis

In a proactive management, managers should anticipate future problems and attack them as soon as possible. To achieve proactive QoS management, QoS analysis tasks must be performed.

Cataloged historical behavior can show, for example, the number of refused RSVP sessions due to lack of resources. If the number of refused sessions increases too much, it indicates that the manager should update network resources. Analysis of a QoS-dependent monitored client/server operation could show the frequency of QoS degradation, and the frequent degradation points. In this case, the manager should check the critical point to see link problems.

One crucial part of QoS analysis is QoS visualization. We consider QoS visualization such an important procedure that it is defined separately from QoS analysis, although it is part of such analysis.

2.6 QoS visualization

Today's network management platforms are topology-oriented, i.e., they show information from a network topological perspective. Managers browse network topology and check each desired device. Some facilities can be found, allowing managers, for example, to ask the platform for a map listing every printer.

For QoS management tasks, current visualization is poor. Managers often search for each important device in maps, and check to see if such device is QoS-enabled. This is a time-consuming task, and should be replaced with an automated search procedure.

QoS visualization is not a task itself. Rather, it is a feature that helps managers to proceed with QoS tasks. Tools should provide visualization based on QoS characteristics. We list here some helpful QoS visualizations.

- **Colored link utilization**. Each link shows, instead of a single black connecting line, a colored set of lines describing link utilization by each flow/aggregate (figure 1, left);
- **Selected end-to-end sessions**. A topology map should have selected end-to-end sessions highlighted through colored lines. Managers could ask to visualize only sessions that match some pre-determined features;
- **QoS enabled devices**. A topology map should highlight QoS-enabled devices. Different colors show devices that implement different solutions. For example, green boxes indicate DiffServ-enabled routers, whereas red boxes indicate RSVP-enabled ones;
- **Segments with QoS degradation**. Segments with QoS degradation could be shown in red or be highlighted (figure 1, right), to indicate degradation. Orange segments could indicate probable degradation coming.

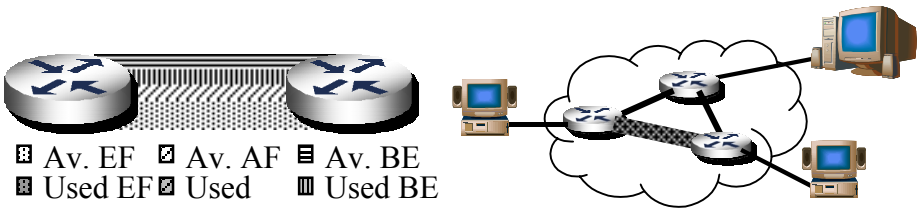

Fig. 1. QoS visualization examples

Several other visualization facilities could be created to help managers identify QoS-related information. Today, QoS visualization can only be found on separate software that has no integration. The next session shows some current solutions used in QoS visualization and other tasks.

3 QAME

Previous sessions presented QoS-related management tasks and some solutions that help managers perform such tasks. Each solution provides functionalities to attack a particular problem, but that is done independently from other solutions. Thus, there is not any QoS task integration.

In addition to the lack of integration, current network management platforms are device-oriented and not QoS-aware, i.e., even if they have QoS support they do not show QoS information properly. A more appropriate environment should allow a network-oriented view to the management, also allowing explicit QoS information.

This session presents the current state in the development of a QoS-integrated management environment, called QAME. The environment is QoS-aware in the sense that it takes QoS information explicitly and shows that information more properly. Besides, QAME provides support for managers to proceed with the six QoS tasks previously defined (QoS installation, operation maintenance, discovery, monitoring, analysis and visualization).

3.1 QAME architecture

QAME architecture extends the policy-based solution defined in [12] by introducing some new elements. The architecture is initially divided into active elements and databases. Active elements perform management and QoS provisioning tasks. They also store, retrieve and update information in databases. Active elements are sub-divided in an upper element (the User Environment), intermediate elements (Policy Consumer, QoS Monitor and Target Finder) and lower elements (Targets). Figure 2 shows QAME elements and databases, which, on their turn, are presented in the following sub-sessions.

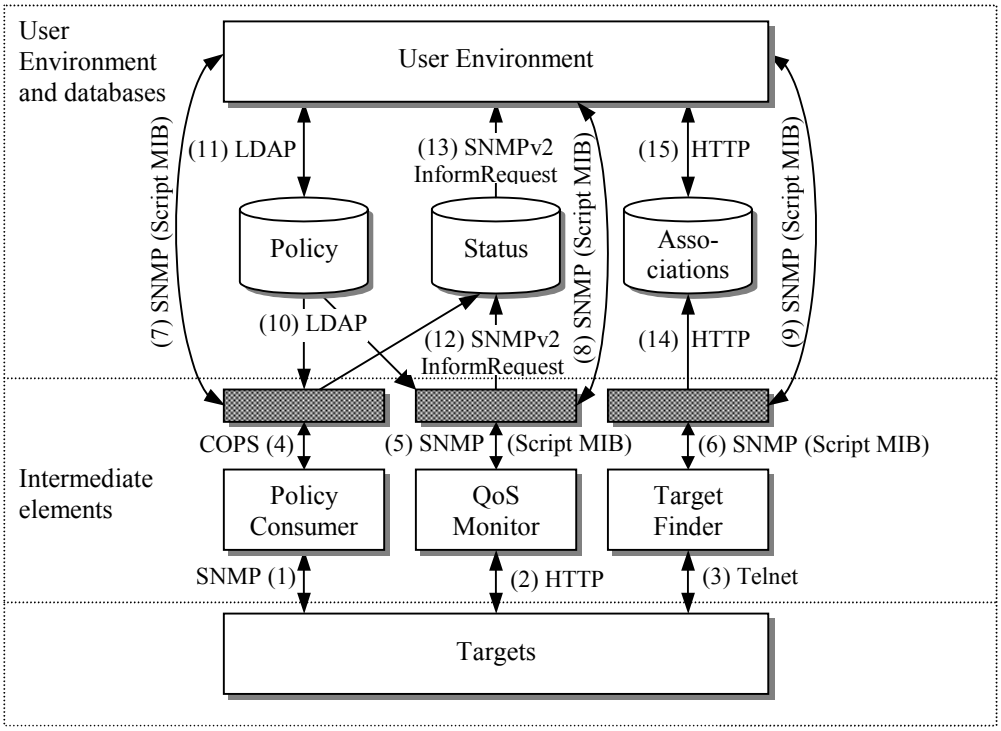

Fig. 2. QAME architecture

Targets
Targets are active elements that influence the final QoS observed in the network. Each device can have several Targets that can influence the network parameters. For example, in a router each interface is a Target. Targets are the final elements that effectively implement a QoS architecture.

Network manager has indirect access to Targets throughout Policy Consumer, QoS Monitor and Target Finder elements. The interface between these elements and Targets is device-specific, and different protocols should be used to access different Targets. A router from vendor A, for instance, can be accessed via a Telnet client, while another router from vendor B can be accessed via HTTP.

Target Finder
In the network, searching each device to identify its Targets is a time-consuming task. Also, new devices just attached to the network must have their Targets cataloged for use. Finally, if

QoS discovery is an important task, automatic Target finding will be necessary.

Target Finders are special agents that search the network for current and new Targets. Each Target Finder recognizes at least one specific Target, using a specific Target finding algorithm. For example, a DiffServ Target Finder is the one that looks within routers and checks the existence of packet prioritization based on the IP DS field. To do that, the DiffServ Target Finder can open a Telnet session or check for DiffServ MIB implementation.

Since each Target can have different capabilities, Target Finders are also responsible for classifying new discovered Targets. Target Finders store any Target information on the Associations database.

Policy Consumer

Policy Consumer is responsible for installing a policy into Targets. Each Policy Consumer, when ordered to install a policy, retrieves the policy by accessing the Policy database. The new policy is then translated into device-specific instructions to program Target to conform that policy.

After policy installation, Policy Consumer is also responsible for checking the success of the policy in the Targets. If a policy could not be installed either due to failure in the Target or to lack of Target capabilities, the Policy Consumer notifies the network manager by sending messages to the User Environment.

A Policy Consumer can install policies in several Targets at the same time. On the other hand, each Target can be associated to several Policy Consumers, even though only one can be the current active consumer.

QoS Monitor

Installed policies might not behave as stated in the policy definition. The QoS resulted from a policy installation can be different from its specification. Critical policies must then have their expected QoS monitored. The element responsible for doing that is the QoS Monitor.

The network manager defines which policies must be checked and QoS Monitors are then associated to the Targets that implement those policies. QoS Monitors access policy definitions also in the Policy database and compare the effective behavior on the network with the one defined in the policy. If degradation is observed, QoS Monitor notifies the network manager by sending special messages to the User Environment, too.

The greater the number of QoS Monitors used, the more accurate the monitoring process will be. Also, the greater the number of QoS Monitors used, the greater the information analysis overhead will be.

User Environment

QAME graphic user interface is implemented in the User Environment, which uses Web technology to show management information. User Environment is responsible for running analysis processes that complement the functionality presented in the Policy Consumer, Target Finder and QoS Monitor. For example, User Environment receives special messages from Policy Consumer telling a policy could not be installed, and messages from QoS Monitor when the observed QoS is different from the expected QoS.

User Environment also interacts with the three databases in order to define their contents. Users define policies that are stored in the Policy database by using the environment interface. Policies can also be modified or removed from the database. Network topology is shown by accessing the Associations database information. Users check network topology on their Web browsers and order actions to be installed in Targets.

Databases

The three databases shown in figure 2 are defined to store different classes of information. Every Target, Policy Consumer, QoS Monitor and Target Finder is registered in the Associations database. With appropriate searching of the base, the User Environment can derive the network topology, existing QoS solutions, and resource capabilities. Furthermore, the Associations database stores the associations between Targets and the other elements. For example, a Target named A uses Policy Consumer B for policy translation and installation, QoS Monitor B for policy performance checking and Target Finder C for discovering possible new capabilities in Target A.

The policies are stored in the Policy database. Since policies themselves do not define on which Targets they should be installed, policies can be stored separately from Targets. We do that because the database needed for Targets and the database needed for policies have different requirements. Policies once defined have little or no change. On the other hand, Targets can have their capabilities extended, associations updated, and have more changeable data overall.

Even more changeable are data used to represent the status of a deployed and/or monitored policy. QoS Monitors and Policy Consumers change data in the Status database every time a deployed policy has its status altered. Thus, Status database binds information stored in the Policy database (the policies) with that stored in the Associations database (the Targets), and introduces new information about this relationship: the status of a policy.

3.2 Elements location

The previous sub-session described each element of the QAME architecture. This present sub-session explains where in the network infrastructures these elements are located and how many elements of the same type can be used.

The more obvious element location is the Target location. Targets are located within network devices that play any active role in QoS provisioning. Target examples are routers and switch interfaces in their queue disciplines. Marking, policing and shaping processes are also examples of Targets. Targets can be located in hosts, too. RSVP-enabled applications, or DiffServ marking processes in end systems [13] are Targets, since they influence the end-to-end QoS.

We leave intermediate elements location for the next paragraphs, since this is a more complicated issue. On the other hand, User Environment location is almost as obvious as Target location was. We use a central point that runs QoS analysis processes and generates HTML pages showing the results. A Web server is also used to allow remote access to the environment. The central point could generate too much traffic on larger networks. Distributed User Environment can be used since databases are detached from the User Environment. Several environments would access the same databases.

Databases can be located on the same device that implements User Environment, or on separate devices. Since there are three databases, some can be found together with User Environment, and others separately. Although figure 2 shows only one copy of each database, for security reasons we could have more copies of the same base and use database replication for consistency. Also, more copies of the same database would facilitate the distribution of network traffic generated by QoS Monitors, Policy Consumers and Target Finders when they need to update databases information.

A trickier aspect is the location of Target Finders, QoS Monitors and Policy Consumers. First of all, since they are independent elements they can be located in different places. QoS Monitors are very tightly related to their Targets. Thus, it is expected that QoS Monitors are

located within the same devices that contain the monitored Targets. However, depending on the installation of the QoS Monitors, they can also be located close to devices, but not inside. For example, a monitor created to check the bandwidth traffic of a router interface could access the MIB-II interface group and realize that an interface is facing overflow, even though the monitor is not located within the router.

Policy Consumers are often located outside devices, but modern equipment is expected to have built-in policy consumer implementations. Even more, Policy Consumers can be located together with the User Environment, thus improving User Environment and Policy Consumer communication.

Finally, Target Finders are often located together with User Environment, acting as special plug-ins that search the network for QoS-enabled devices. Target Finders can also be located in network segments other than the User Environment segment. They would act as segment monitors, looking for QoS traffic generated by QoS-enabled devices. The less suitable location for a Target Finder is within devices, since devices and their Targets are the objects of the finding process. One exception is when devices are able to announce themselves and their capabilities to the network, registering on the databases. Up to now, authors are unaware about devices with this feature.

Table 1. QAME elements location. Rows list QAME elements and colums list possible locations. Cells marked with an "x" denote that the QAME element in the row can be present in the equipment of the colum. "Devices" are network equipment (routers, switches, bridges, etc.). "Proxies" are network equipment used to host some active elements that act on different equipment (e.g., a QoS Monitor located within a host used to monitor a router). "Hosts" are listed to explicitly define elements located and acting in a host. Finally, "management stations" are used to denote the hosts where QAME User Environment and databases are placed.

	Devices	**Proxies**	**Hosts**	**Management stations**
Targets	x	--	x	Only if target plays active role in QoS provisioning
QoS Monitors	x	x	x	x
Policy Consumers	x	x	x	x
Target Finders	--	x	--	x
User Environment	--	--	--	x
Databases	--	--	--	x

One important issue about location is the management interface between the User Environment and Target Finder, QoS Monitor and Policy Consumer. This interface is depicted in figure 2 by the gray rectangles connecting User Environment and intermediate elements. These interfaces can be found together with the elements if the elements implement such interfaces. Otherwise, the interfaces are located in the User Environment and translate requests into element-specific commands. This separation between element implementation and interface is important because modern devices could implement elements with interfaces different from those used by QAME. In this case an interface translation is needed to allow the use of built-in device elements. Table 1 summarizes the possible location of QAME elements.

3.3 Protocols

This sub-session describes the protocols used to provide communication between QAME elements. In the protocol definition phase of the project, we decided to use standard and open protocols to implement such communication. Even though some standard protocols might not be the best choice for some critical tasks, we believe that this choice makes our architecture open, making future implementation of new modules easier.

Targets protocols
The protocols used to communicate with Targets are actually defined by the devices that contain the Targets. These protocols are then dependent on the device's provider, which could choose to use standard protocols or implement its own proprietary protocol.

The most common Targets protocols available are Telnet and SNMP, but modern devices also use HTTP for configuration management (figure 2, labels 1, 2 and 3). Fortunately, proprietary protocols are more rare.

Despite the diversity of possibilities, Targets protocols are not a critical issue since intermediate elements are responsible for protocol translations. Thus, when two different routers, using different protocols, should be programmed to prioritize a defined flow, that programming task "sees" the different routers as equal because of the protocol translation executed in the Policy Consumer. This translation also occurs in the QoS Monitor and Target Finder elements (figure 2, labels 1 and 4, 2 and 5, 3 and 6).

QoS Monitors, Policy Consumers and Target Finders protocols
Protocols used to communicate with intermediate elements can be divided into two groups: those implemented by the elements, and those used as interface with elements (gray rectangles in figure 2). Protocols implemented by the elements are element-dependent and defined by the element developer. For example, a vendor implementing a Policy Consumer within its new router could use COPS [14] to transfer policies. Another vendor could choose Script MIB [15] definitions to have access to the policies (figure 2, labels 4, 5 and 6).

On the other hand, protocols used in the access interface are always the same. This is a requirement to allow access to the same interface in the User Environment. Thus, interface actually implements only protocol translation, from User Environment interface access into the intermediate elements-specific interface (figure 2, labels 4 and 7, 5 and 8, 6 and 9). The current QAME implementation uses Script MIB to communicate with intermediate elements interface.

Database protocols
The protocols used to access database information are different because the nature of each base is different. Policy database is reached using an LDAP [16] protocol (figure 2, labels 10 and 11). Since policies are information that has few updates but can be accessed several times, a write-once read-several times protocol like LDAP is more suitable.

Status and association information are more dynamic, and LDAP protocol should be avoided. The current implementation uses SNMPv2 InformRequest message to update status information in the Status database (figure 2, label 12). InformRequest messages can be faced as an SNMPv1 trap message with confirmation. Thus, QoS Monitor perceiving QoS degradation can update the status of a monitored flow or aggregate trapping the Status database, and still have confirmation of the update operation due to the reply of the InformRequest message. Policy Consumers can also notify the Status database when a policy deployment fails. The InformRequest message can be forwarded to the User Environment

when critical monitoring tasks are performed (figure 2, label 13). Finally, User Environment and Target Finders reach Associations database through HTTP and queries to a PHP4 engine [17] (figure 2, labels 14 and 15).

4. Conclusions and future works

Providing QoS services in networks is currently very important because time-dependent application cannot be deployed using best-effort based services. QoS architectures must be installed, but should also be properly managed to be effective. Traditional network management cannot be applied to QoS management because the amount of available information is much larger and complex.

Current efforts try to find a way to allow complexity abstraction by using policy-based management. But policies are not sufficient to allow total QoS management. Other QoS management-related tasks are performed by network managers, and support should be available for those tasks.

This paper has defined six main QoS management related tasks: QoS installation, operation maintenance, discovery, monitoring, analysis and visualization. Policy-based management only addresses QoS operation maintenance. Other tasks have no advantage of policy-based management. For these tasks we have presented current solutions that help managers install them. Unfortunately, there is no integration between solutions, and managers often have to deal with too many tools, increasing the management complexity.

To make management easier we have built a QoS management environment called QAME where the six important QoS tasks can be performed in an integrated fashion. We have described the QAME architecture, its elements, and where these elements can be located in the network (e.g., within routers, switches, hosts, or integrated in the management station). QAME elements exchange information to allow integration. This information exchange is done by communication protocols. We have shown that the QAME environment uses LDAP, SNMPv2, Script MIB and proprietary protocols to implement communications.

Protocol translation is a required feature to allow better interaction between user environment (where management information is presented) and lower-level elements (where management information is gathered and processed). The protocol translation is possible because QAME architecture detaches lower-level elements implementation from their interface. The element implementation can be located, for example, in a router, while the element interface can be located in the management station.

Future works may address security, database replication and distributed management issues. Since SNMP messages are not encrypted, the use of SNMPv3 is a natural choice. Database replication deserves a more accurate research because single database instances could be inadequate to manage very large networks. Because management traffic will be greater, database access at a single point could prevent better management performance. Also, in larger networks, a central point of management should be avoided, and distributed management with more than one User Environment would be preferable.

References

1. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. Internet draft <draft-ietf-mpls-arch-07.txt>. Work in progress (2000)
2. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services. Request for Comments 2475 (1998)

3. Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. Request for Comments 2205 (1997)
4. Shenker, S., Wroclawski, J.: General Characterization Parameters for Integrated Service Network Elements. Request for Comments 2215 (1997)
5. Nichols, K., Jacobson, V., Zhang, L.: A Two-bit Differentiated Services Architecture for the Internet. Request for Comments 2638 (1999)
6. Case, J., Fedor, M., Schoffstall, M., Davin, J.: A Simple Network Management Protocol (SNMP). STD 15, Request for Comments 1157 (1992)
7. McBride. D.: The SLA Cookbook: A Recipe for Understanding System and Network Resource Demands. Hewlett-Packard Company. Available via URL http://www.hp.com/openview/rpm (1996)
8. Yavatkar, R., Pendarakis, D., Guerin, R.: A Framework for Policy-Based Admission Control. Request for Comments 2753 (2000)
9. Fenner, W.: Internet Group Management Protocol, Version 2. Request for Comments 2236 (1997)
10. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. Request for Comment 1889 (1996)
11. Jiang, Y., Tham, C.K., Ko, C.C.: Providing Quality of Service Monitoring: Challenges and Approaches. NOMS 2000 - IEEE/IFIP Network Operations and Management Seminar (2000)
12. Mahon, H., Bernet, Y., Herzog, S.: Requirements for a Policy Management System. Internet draft <draft-ietf-policy-req-02.txt>. Work in progress (2000)
13. Granville, L., Uzun, R., Tarouco, L., Almeida, J.: Managing Differentiated Services QoS in End Systems using SNMP. IPOM 2000 - IEEE Workshop on IP-oriented Operations & Management (2000)
14. Chan, K., Durham, D., Gai, S., Herzog, S., McCloghrie, K., Reichmeyer, F., Seligson, J., Smith, A., Yavatkar, R.: COPS Usage for Policy Provisioning. Internet Draft <draft-ietf- rap-cops-pr-02.txt>. Work in progress (2000)
15. Quittek, J., Kappler, C.: Remote Service Deployment on Programmable Switches with the IETF SNMP Script MIB. DSOM 2000 – IFIP/IEEE International Workshop on Distributed Systems: Operations and Management. Springer Werlag (1999)
16. Arlein, R., Freire, J., Gehani, N., Lieuwen, D., Ordille, J.: Making LDAP active with the LTAP gateway: Case study in providing telecom integration and enhanced services. In Proc. Workshop on Databases in Telecommunication (1999)
17. Hypertext Preprocessor – PHP4. Available via WWW at URL: http://www.php.net (2001)

An Approach for Integrated Management of Networks with Quality of Service Support Using QAME

Lisandro Zambenedetti Granville,
Márcio Bartz Ceccon, Liane Margarida Rockenbach Tarouco,
Maria Janilce Bosquiroli Almeida and Alexandre da Silva Carissimi

Federal University of Rio Grande do Sul - UFRGS
Institute of Informatics - II
Av. Bento Gonçalves, 9500 - Bloco IV
Porto Alegre,RS - Brazil
{*granville, ceccon, liane, janilce, asc*}*@inf.ufrgs.br*

Providing QoS-guaranteed services in current installed networks is an important issue, but only deploying QoS services is not enough to guarantee their success: QoS management must also be provided. Nowadays, policy-based network management (PBNM) addresses this need, but such management is not enough either. Network managers deal with QoS tasks that cannot be performed using only PBNM. Other solutions, besides PBNM, have to be used to proceed with QoS management-related tasks. Unfortunately, these solutions are independent from each other, leading to a scenario where integration is difficult. This paper introduces QAME (QoS-Aware Management Environment) which main goal is the provisioning of facilities to allow a common and integrated Web-based management of QoS-enabled networks.

Keywords: QoS management, policy-based network management, Web-based management environment

1 Introduction

The great majority of today's networks operate based on the IP best-effort approach. There is no warranty concerning traffic delay, jitter, throughput or lost rate. Several applications operate properly in this environment, but several others can only be delivered if network QoS (Quality of Service) warranties are present [1].

Managers should be aware of QoS features present on the network. QoS architectures can only be effective and provide guaranteed services if QoS elements are adequately configured and monitored. Thus, in addition to the management of traditional elements (routers, switches, hosts, etc.), managers must also manage QoS aspects. In this scenario, it is not a surprise to realize that the overall management will be more complex [2].

QoS management must take place within the same environment used to manage standard network elements and management platforms should be aware of QoS. A common environment is required to allow managers to proceed with QoS and standard management-related tasks in an integrated fashion [3]. Management platforms should be aware of QoS to facilitate the visualization and manipulation of QoS information, at least as easy as it is currently possible dealing with standard management information (e.g. routing tables, interface monitoring, etc.). Unfortunately, today's network management platforms are not QoS-aware and managers are forced to investigate

each QoS-enabled device to check QoS parameters. Thus, a network-oriented approach that complements current device-oriented management is needed, with features that explicitly present QoS information to managers in an intuitive fashion [2].

In this context, we propose a Web-based management environment able to proceed with both standard and QoS management-related tasks. We introduce QAME (QoS-Aware Management Environment) which is based on the IETF PBNM works, and uses open standards (e.g. SNMP, LDAP, COPS, ScriptMIB) to proceed with management tasks. Managers first ask for QAME topology and QoS discovery services to map the environment to be managed. Then, policies are defined and scheduled to be used in specific devices. QAME QoS monitors can be activated to check critical network flows/aggregates in the most important network segments, while analysis and visualization features show information about experienced and expected network QoS.

Since QAME is a Web-based environment, the graphical user interface is the manager's favorite Web browser. QAME is built with PHP4 script engine that allows an extension of the environment "on the fly". New modules can be easily added throughout an upload mechanism, without having the QAME execution to be stopped. We have used XML to describe both management information and presentation data. XSL transformation, on server-side, is used to allow the development of further user interfaces, besides QAME standard user interface. Intermediate scripts are responsible for other transformations (e.g. from MIB to XML and from LDAP to XML).

The main contribution of our proposal is that QAME allows the execution of QoS management-related tasks in an integrated fashion that is absent in current management platforms. Another benefit is that QoS-related information is shown by the environment in more explicit ways, easing QoS visualization and QoS-related information treatment. Also, managers that require the incorporation of more specific features, not found in the original environment, can easily extend QAME using open standard frameworks (e.g. XML) and free software engines (e.g. PHP).

This paper is divided as follows. Section 2 discusses related work about QoS and Web-based management. Section 3 presents QAME architecture describing its components and the elements location in the managed network. QAME implementation and associated technologies are presented in section 4, while section 5 explains QAME operations throughout an example. Finally, section 6 concludes the paper and also shows future work to be considered.

2 Related Work

Solutions created to manage QoS in modern networks gained recent highlight mainly because of the PBNM proposals. The main players in the IP network management market issued their PBNM solutions, often integrated in their standard management platforms. Hewllett-Packard created its PolicyXpert [4] that is part of the HP OpenView management suite. Cisco also produced its PBNM solution and released, in 1999, the QPM (QoS Policy Manager) as part of the CiscoAssure policy management initiative [5]. Extreme Networks did the same and created the EPICenter (formerly the ExtremeAware Enterprize Manager - EEM) [6]. Other industry players have issued their solutions too (e.g. Nortel Networks, Lucent Technologies, Orchestream) [7]. Although these solutions can have market success, they suffer from a lack of integration from each other. Since current PBNM systems are not entirely based on IETF open standards, costumers will face serious problems when different solutions will start to be used in the same managed network [8].

In research areas that investigate QoS management some important work can be found. Imperial College London have been an active PBNM research player, with pioneer efforts, mainly through Emil Lupu and Morris Sloman works [9]. Mahon et. al. are working on the definition of general requirements for PBNM systems [10]. In their IETF work, a general architecture has been defined to guide the development of PBNM solutions. However, as Mahon states, that proposals don't address other critical aspects of QoS management, such as QoS monitoring, analysis and discovery.

Such other QoS management aspects are addressed, on the other hand, by other research projects. Hong et al. [11] proposed, in 1998, a CORBA-based management framework for managing QoS of distributed multimedia services and applications of the MAESTRO system. The

layered architecture enabled an end-to-end management of QoS provisioning resources, including services for QoS specification and mapping, admission control, negotiation and renegotiation. A generic QoS MIB was developed to access QoS parameters in such layered architecture. In 2000, we have also proposed SNMP as a mean to program DiffServ marking processes on end-systems and to allow users to reserve QoS resources asking for such resources to Bandwidth Brokers (BB) [12]. QoS monitoring has been studied by Jiang et al. [13], and Joshi et al. [14] presented, also in 2000, a solution that integrates traditional network management and QoS monitoring.

All previous proposals and solutions are important for QoS management, but they are independent from each other. In this scenario, network managers are forced to deal with several tools to manage different aspects of the whole QoS issue. Recently, we have classified QoS management-related tasks [15], trying to organize, from a network management point-of-view, the different aspects involved in QoS management. We have divided QoS management in six different tasks (installation, operation maintenance, discovery, monitoring, analysis and visualization) and have argued that an effective QoS management system is the one able to provide, in an integrated environment, facilities to proceed with the execution of such tasks. Also, we have argued that Web-based management should also be applied to QoS related tasks, in a way to allow network managers to control QoS aspects using Web widely known facilities.

Web-based management has been investigated for some years, but its first significant move occurred in 1996, when WBEM (Web-Based Enterprize Management) first came to life [16] as part of the DMTF (Distributed Management Task Force) solution. Martin-Flatin et al. have also done investigation on Web-based management and proposed the JAMAP management platform, based mainly on Java technology [17]. Using Java Applets, managers can interact with the managed network and the associated management structures.

XML technology used on network management found its way when XML was incorporated to the WBEM framework. From a different approach, John et al. presented, in 1999, XNAMI [18], that uses XML to reduce management traffic allowing the transfer of MIBs between managed devices and management stations. Unfortunately, none of the previous Web-based proposals took QoS into account directly, i.e. QoS management were not the main focus.

3 QAME architecture

QAME architecture (fig. 1) is divided in three different sets of elements: upper elements (which include manager Web browser, the Web-based user interface and databases), intermediate elements (policy consumers, QoS monitors and target finders) and lower elements (targets). Three different databases are part of the QAME architecture's upper elements: policy, status and associations databases.

3.1 Targets

Targets are active elements that influence the final QoS observed in the network. Each device can have several targets that influence in the QoS provisioning. For example, in a router, each interface's queueing discipline is a target. Thus, targets are the final elements that effectively implement a QoS architecture.

Managers access the networks' targets indirectly throughout QAME intermediate elements (policy consumers, QoS monitors and target finders). The interface between these elements and targets depends on the devices that own the targets. Here, different protocols have to be used to access different targets on different devices. A router from vendor A, for example, can have its interfaces programmed via Telnet, while another router from vendor B requires HTTP interactions to change its interfaces settings.

3.2 Policy consumer

Policy consumer is the element responsible for the installation of management policies into targets. When ordered (label 1), the policy consumer retrieves policies definitions from the policy

database (label 2). These policies are then translated into device-specific instructions to program the appropriate targets to conform the policies definitions (label 3).

After policy installation, the policy consumer is also responsible for checking the success of the policy deployment. If a policy can not be installed, either due to a failure in the target or to a lack of target capabilities, the policy consumer notifies the network manager by sending messages to the user environment (label 1).

The status database stores the deployment and operating status of each active policy associated to targets. Failures in policy deployment make the policy consumer update the status database indicating the installation problem (label 4).

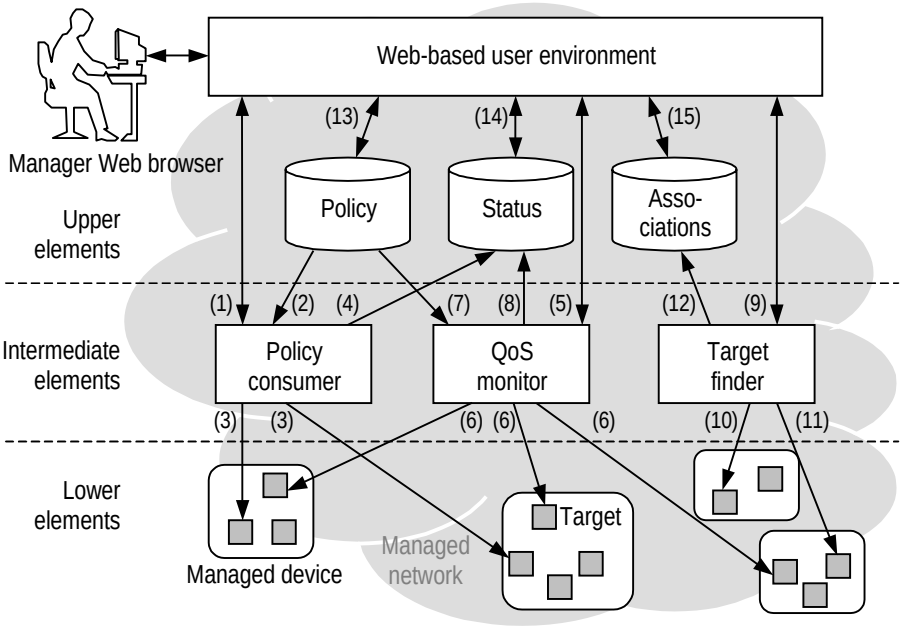

Fig. 1: QAME architecture

3.3 QoS monitor

Installed policies may not behave as stated in the policy definition. The QoS resulted from a policy installation can be different from its specification. Critical policies must then have their expected QoS monitored. The element responsible for doing that is the QoS monitor.

The network manager defines which critical segments must be checked (label 5) and QoS monitors are then activated to the check the targets that make part of those segments (label 6). QoS monitors access policy definitions also in the policy database (label 7) and compare the observed behavior of the network with the one defined in the policy. If degradation is verified, QoS monitor notifies the network manager by sending special messages to the user environment (label 5) and updates the status database to be consistent with the problem observed (label 8).

3.4 Target finder

In the network, searching each device to identify its targets is a time-consuming task. Also, new devices just attached to the network must have their targets identified for future programming. To do that, target finder element is the one trigged (label 9) to run QoS discovery services.

Target finders search the network for current and new targets. Each target finder recognizes at least one specific target using a specific target finding algorithm and a specific device access protocol. For example, a DiffServ target finder is the one that looks within routers and checks the

existence of packet prioritization based on the IP DS field. To do that, the DiffServ target finder can open a Telnet session (label 10) or check for target DiffServ MIB implementations (label 11).

Every discovery target is identified, classified and stored in the associations database (label 12). Target's device is also stored and relations between targets and their corresponding devices are created. Target finders are responsible for coordinating all this procedures.

3.5 Web-based user environment

QAME graphical user interface is implemented in the Web-based user environment, which uses Web technology to show management information. User environment is responsible for running analysis processes that complement the functionality presented in the policy consumer, QoS monitor and target finder. For example, user environment receives special messages from policy consumer telling that a policy could not be installed (label 1), and messages from QoS monitor when the observed QoS is different from the expected QoS (label 5).

User environment also interacts with the three databases in order to define their contents. Users define policies that are stored in the policy database (label 13). Policies can also be modified or removed from the database. Network manager can check the status of a deployed policy accessing the status database (label 14) and network topology is shown by accessing the associations database information (label 15).

3.6 Elements location

The previous subsections described each element of the QAME architecture. The present subsection explains where, in the network infrastructures, these elements are located.

Targets are located within network devices that play any active role in QoS provisioning. Examples of targets are routers and switches interfaces and their queueing disciplines. Marking, policing and traffic shaping processes are also examples of targets. Targets can be located in hosts, too. RSVP-enabled applications, or DiffServ marking processes in end systems [12] are targets, since they influence the end-to-end QoS.

Web-based user environment location is almost as obvious as target location was. We use a central point that runs QoS analysis processes and generates, via PHP4 engine, HTML pages showing the results. Databases can be located on the same device that implements Web-based user environment, or on separate devices. Since there are three databases, some can be found together with user environment, and others separately. Although figure 1 shows only one copy of each database, for security reasons we could have more copies of the same base and use database replication for security. Also, more copies of the same database would facilitate the distribution of network traffic generated by policy consumers, QoS monitors and target finders when they need to update databases information.

A trickier aspect in elements location is the location of target finders, QoS monitors and policy consumers. First of all, since they are independent elements they can be located in different places. QoS monitors are very tightly related to their targets. Thus, QoS monitors are expected to be located within the same devices that contain the monitored targets. However, depending on the installation of the QoS monitors, they can also be located close to devices, but not inside. For example, a monitor created to check the bandwidth traffic of a router interface could access the MIB-II interface group and realize that an interface is facing overflow, even though the monitor is not located within the router.

Policy consumers are often located outside devices, but modern equipments are expected to have built-in policy consumers. On the other hand, policy consumers can be located together with the user environment. Finally, target finders are often located together with user environment, acting as special plug-ins that search the network for QoS-enabled devices. Target finders can also be located in network segments other than the user environment. The less suitable location for a target finder is within devices, since devices and their targets are the objects of the finding process.

Table 1 summarizes the possible location of QAME elements.

Tab. 1: QAME elements location. Rows list QAME elements and columns list possible locations. Cells marked with an "x" denote that the QAME element in the row can be present in the equipment of the column. "Devices" are network equipments (routers, switches, bridges, etc.). "Proxies" are network equipment used to host some active elements that act on different equipment (e.g., a QoS monitor located within a host used to monitor a router). "Hosts" are listed to explicitly define elements located and acting in a host. Finally, "management stations" are used to denote the hosts where QAME Web-based user environment and databases are placed.

	Devices	**Proxies**	**Hosts**	**Management stations**
Targets	x	-	x	Only if target plays active role in QoS provisioning
Qos Monitors	x	x	x	x
Policy Consumers	x	x	x	x
Target Finders	-	x	-	x
User Environment	-	-	-	x
Databases	-	-	-	x

4 QAME technologies

In this section we will present the technologies involved in the implementation of the QAME prototype. These technologies are present in two important aspects of the QAME communications: among QAME elements and the Web-based user environment/manager Web browser communication.

4.1 Techonologies of the QAME elements

As stated before, the communication with targets depends on the devices that own the managed targets. Each device can implement different ways to monitor and program its targets. For example, several commercial routers allows programming of their internal structures through Telnet (the standard old method), SNMP (for standard management platforms) and HTTP (for Web-based management). COPS protocol is expected to be found more and more frequently, supporting PBNM in modern equipments.

This diversity of access methods introduces several complexities in targets communications. QAME intermediate elements are then forced to own the following properties when communicating with targets:

- *Intermediate elements know which kind of information they are dealing with.* A policy consumer, for example, knows if an associated target supports DiffServ or IntServ. When a policy is installed, the policy consumer translates the policy definition to device-specific programming (DiffServ or IntServ). When a QoS monitor checks for performance issues, it knows if it must watch for aggregate performance (in the case of DiffServ) or flow performance (in the case of IntServ).

- *Intermediate elements know how to access targets information.* A policy consumer, for example, knows if it must use Telnet session, SNMP or HTTP to program a target. A QoS monitor knows if DiffServ information, in a specific target, is reached with COPS, SNMPv1, v2 or v3. Also, a target finder looking for targets in router from vendor A knows that those targets are found if COPS is used.

Figure 2 depicts three examples of different policy consumers acting in different devices' targets. The devices are routers from three different vendors. The targets within routers A and C are accessed via SNMP, while targets from router C are accessed via COPS. Although routers A and

B support DiffServ, they are accessed with different protocols. It means that any combination of access protocol and supported QoS mechanism requires a different policy consumer. This feature is also valid for QoS monitors and target finders. In our prototype, we have created intermediate elements for DiffServ and IntServ using SNMPv1 and Telnet, i.e we have 4 different policy consumers, 4 different QoS monitors and 4 different target finders.

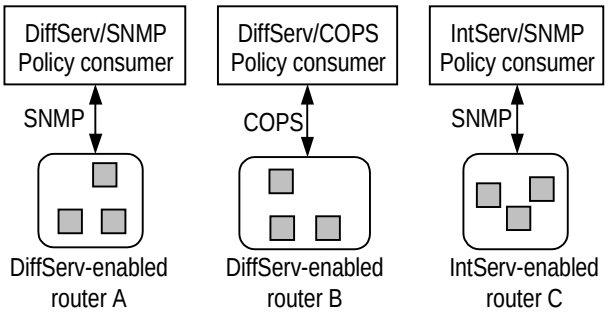

Fig. 2: Targets communication examples

Despite the complex communication with targets, the communication between intermediate elements and the Web-based user environment is simple. We implemented this communication using the Script MIB [19] definitions. When the network manager wants some actions to be executed in the managed network, an appropriate script is selected and sent to an intermediate element (figure 3, labels 1, 2 and 3). Each intermediate element provides facilities that allows the script to deal with the targets to be accessed and the databases to be updated or consulted. We have used the Jasmin implementation to use Script MIB definitions in QAME environment. Figure 3 shows the communication with the QAME intermediate elements and the communication with the databases.

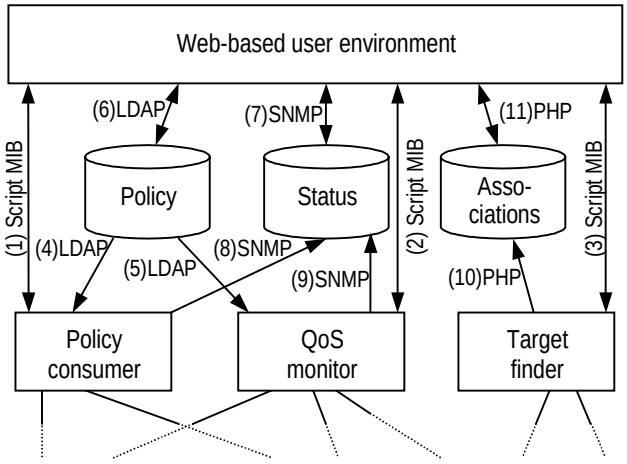

Fig. 3: QAME elements communication

Policy database is implemented with LDAP. We have use OpenLDAP package that includes the LDAP server and protocol API in the implementation of the policy consumers (label 4) and QoS monitors (label 5). For Web-based user environment communication the OpenLDAP API was hidden by the LDAP API provided by the PHP4 engine (label 6).

The status database interface was implemented as a MIB (label 7). We used NET-SNMP package that supports SNMPv3. A key aspect of status database communication is related to the notification messages. We used SNMP InformRequest message to update the status database because the

InformRequest works as a trap message with reply. This allows the policy consumers (label 8) or QoS monitors (label 9) to notify the status database and still be sure that the notification reached the destination.

Finally, the associations database is implemented using the MySQL solution. Thus, intermediate elements act as MySQL clients to access the needed information using PHP4 scripts (label 10). Also, Web-base user environment access the MySQL using the PHP4 (label 11). Although PHP4 has specific MySQL functions, we used the DB class that provides database abstraction, which allows an easier replacement of MySQL, if required in the future.

4.2 QAME Web-based technologies

A key aspect in the QAME communication is the technologies involved in the interaction between the Web-based user environment and the manager Web browser. We have balanced the presentation overhead of the managed network between server and browser. This is done in a way to allow a richer user interaction, without introducing too much network messages exchange.

Web browser is responsible for asking network topology information for the server, and the respective presentation. We use Flash technology to build the network topology in the browser. The first advantage of using Flash is that the network traffic is reduced, since no pictures are passed from server to browser; only presentation information is exchanged. Second, Flash allows an effective user interaction, where network manager can change topology layout directly on the Web browser.

Flash itself does not provide all the facilities required in the browser interaction. JavaScript technology is then applied to complement the Flash presentation. Figure 4 shows an example of a network presented in the QAME user interface using the Flash facilities. The figure show QAME menu on the right-hand side and a topology built with Flash on the left. Each managed element has a particular set of management operations that can be applied to. This operations can be accessed through the context-menus (also shown in the figure 4) trigged when user clicks some managed device with the CTRL key pressed.

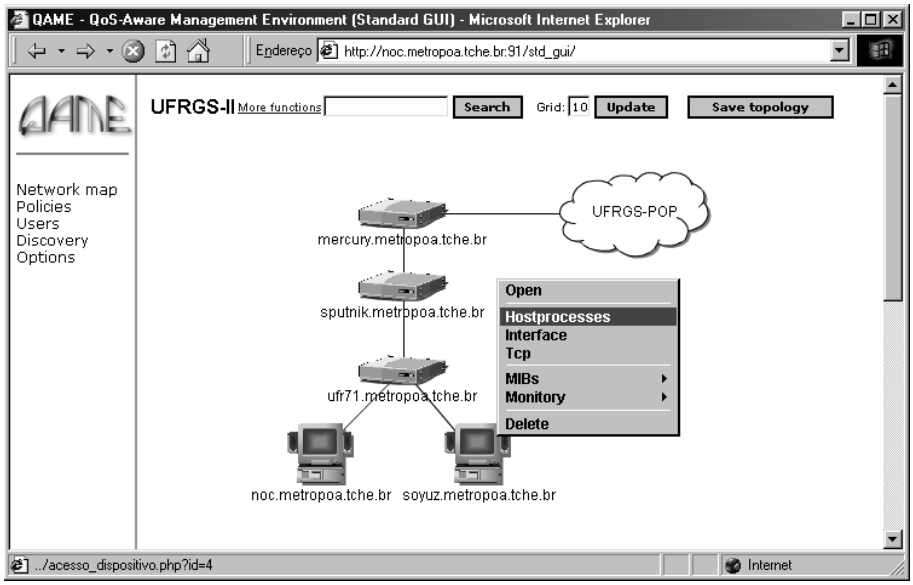

Fig. 4: Network topology presentation in QAME user environment

In the server-side, we use PHP4 engine, as mentioned before, and XML translations. PHP4 orchestrates several tasks, including databases access, intermediate elements communication and

user-environment graphical presentation. XML, on its turn, is used to present several information in a standard way. Table 2 shows examples of PHP4 scripts that retrieve information from different sources and produce XML results. Frequently, the source accessed does not exposes its data in XML, leaving to PHP4 the task to translate from a source representation (e.g. MIB) to XML. In the case the source already exposes its data using XML, the scripts only copy the results to the user-environment.

Tab. 2: Examples of PHP4 scripts that translates information from different sources to XML.

PHP4 Script	**Source**	**Output**	**Description**
route.php?ip=w.x.y.z	hosts and routers	route.xml	Retrieves a devices' route table accessing the ipRouteTable objects from the 'w.x.y.z' device
procs.php?ip=w.x.y.z	hosts	procs.xml	Retrieves the list of processes from the host 'w.x.y.z' as specified in the HostResources MIB
policy.php?pol=p	policy database	policy.xml	Retrieves the policy 'p' from the policy database
status.php?pol=p.t.w.x.y.z	status database	status.xml	Retrieves the operational status of policy 'p' applied to the target 't' within device 'w.x.y.z'
targets.php?ip=w.x.y.z	association database	targets.xml	Retrieves the targets from device 'w.x.y.z'

When the resulted XML is about to be sent to the manager Web browser, we use XSL to transform the original XML to a HTML standard page. We use Sablotron solution to proceed with the XML/XSL transformation in the server-side, allowing not XML-enabled browsers to access the retrieved information. One XML file can be differently transformed using different XSL files. With this feature we implemented, until now, three different QAME skins: advanced skin, standard skin and text only. Advanced skin produces HTML pages with several graphical elements and is preferable used when the manager operates his/her Web browser in the local network, near to the Web server. The standard skin produces fewer graphical elements and should be used when manager operates far away from the managed network. Although, if the communication between manager Web browser and QAME Web server is too congested, the text only skin can be used, since almost no graphical information is generated. The desired skin is selected when manager logins into the QAME environment.

Table 3 summarizes the technologies used in the current QAME prototype implementation.

Tab. 3: QAME technologies.

Technology	**Interaction**
Telnet, SNMPv1	Used to access DiffServ and IntServ-enabled devices
Script MIB	Script MIB is the interface of the intermediate elements
LDAP	Used to access the policy database
SNMPv2(v3) MIB	Implements the access to the status database
PHP + MySQL	Used to access the associations database
Flash + JavaScript	Dynamic topology interaction on Web browser
PHP + XML	Translates information from different sources to XML
XML + XSL	Implements the QAME skins feature
Sablotron	Transforms XML/XSL to HTML in server-side

5 An example of QAME usage

In this section we will present an example of how QAME can be used to proceed with a simple set of management tasks. In this example we are dealing with a network composed by two connected segments that support DiffServ in their border routers (figure 5). This network is not yet mapped, an thus QAME knows nothing about the network to be managed, expect the IP addresses of its intermediate elements. Two policy consumers are used: one is implemented within the router belonging to the right-hand segment and the other is implemented by a host that controls the other router from the left-hand segment. Just one QoS monitor and one target finder are applied. The network experience hard HTTP traffic between the two segments. This hard traffic must be controlled to allow the videoconferencing to run between the hashed hosts in the figure 5.

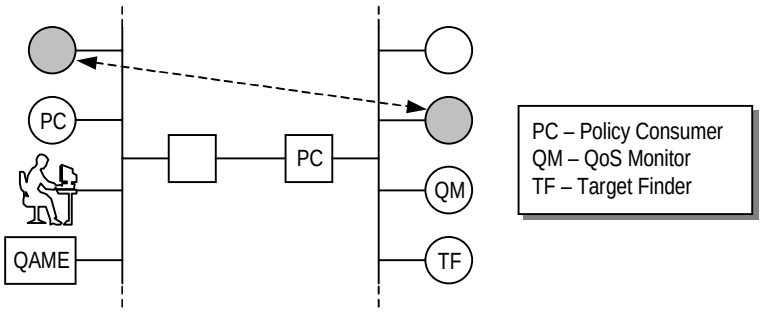

Fig. 5: A network example

The first step is the definition of a discovery rule. This rule is divided in topology discovery and QoS discovery. Figure 6a shows a rule defined to search the network 143.54.0.0 with netmask 255.255.0.0. We will search for DiffServ-enabled devices. The rule will be activated "now" and reactivated every 24 hours, until December 24, 2003. More specific discovery parameters, although not shown in figure 6a, can also be specified (for example, the number of ICMP retransmission in case of errors and the timeout of ICMP replies).

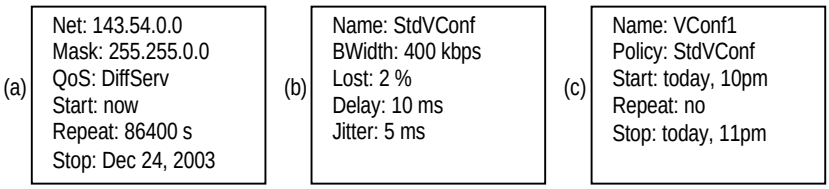

Fig. 6: Examples of discovery rule, policy definition and deployment

QAME transfers the discovery rule to the appropriate target finder based on the IP address, netmask and QoS mechanism specified in the rule, although in our example there is no choice since just one DiffServ target finder exists. The rule is kept in the target finder until its end (December 24, 2003) and started just after its transfer. The target finder than first discover the network topology using ICMP messages. The just discovered devices are then investigated, in our current prototype via SNMP, to determine its DiffServ capabilities. Then, target finder stores the discovered devices and the associated targets in the associations database that is located together with QAME. Visually, network manager can perceives the evolution of the discovery by looking at the Flash topology built in the user browser. Since Flash updates its presentation periodically, new discovery devices are shown in next updates.

The next step requires the manager to determine some relationships among QAME elements and the managed network. The QoS monitor, from the right-hand segment, has to be associated to the right router. The policy consumer from the left-hand segment has to be associated to the

left router. The policy consumer from the right-hand router is automatically associated to the router by the target finder. All associations are then stored in the associations database. After that, network is ready to be programmed.

Policies must be defined. Figure 6b shows the definition of a simple policy used to guarantee enough network resources to the videoconferencing we are dealing with. This policy is stored in the policy database for future use, and indexed by its name. Let's suppose the videoconferencing should gain network resource for only one hour beginning at 10am. Figure 6c shows the deployment rule for this policy stored in the status database and indexed by its name too.

To deploy the policy on the network, manager consults QAME topology facilities to determine the routers in the path from one videoconferencing host to the other. These routers are highlighted on the topology presentation and manager is able to deploy the policy. Internally, QAME consults the associations database to retrieve the IP addresses of associated policy consumers for each router. Finally, policy is sent to these consumers that translate the policy and program the network devices when the policy is activated.

To watch the experienced QoS in the network, manager again consults QAME topology facilities to determine which routers can be monitored. In this case, only the router from right-hand segment has a QoS monitor associated. The policy definition and the deployment rule are then accessed by the QoS monitor to verify when monitoring should begin.

6 Conclusion and future work

In this paper we have argued that QoS provisioning architectures can deploy guaranteed services only if such architectures are managed by QoS management processes. In the related work section we saw that several aspects involved in QoS management are addressed by different solutions, leading to a scenario where managers are forced to deal with different tools.

The presented QAME solution integrates, in the same environment, QoS management-related tasks that were separately found before. An advantage of this integration can be observed, for example, in policy deployment followed by QoS monitoring. In a standard environment, managers could use HP PolicyXpert to define a policy and deploy it. Then, using a separated monitoring tool the managers proceed with another definition to determine which flows have to be observed. In QAME, however, only one definition is necessary to program the network and to observe the network behavior, since policy consumers and QoS monitors use the same policy definition and deployment rule to proceed with their tasks.

Another important aspect of QAME is that, even though it is a Web-based management environment, the network manager is not limited by static network bitmaps or heavy loaded Java applets. We combined Flash and JavaScript to provide, with low bandwidth usage, levels of user interaction that are normally found only in standard not Web-based platforms.

Future work is related to the definition, in a higher abstract language, of policies to be applied. Today, policy definition still requires several details that could be abstracted from the network manager. More investigation is also required in the topology representation of very large networks (more then 400 devices) using the JavaScript/Flash solution. We have used QAME to manage small (less then 100 devices) and medium networks (between 100 and 400 devices). Its behavior have shown that QAME integration of QoS management facilities eases, in that networks, the management. On very large networks, on the other hand, the amount of information represents a very challenger management task.

References

[1] Stardust.com, Inc.: The Need for QoS. White paper. Available at: <http://www.qosforum.com/white-paper/Nedd_for_QoSv4.pdf>. QoS Forum (1999)

[2] Eder, M., Nag, S.: Service Management Architecture Issues and Review. RFC 3052. IETF (Jan. 2001)

[3] Huston, G.: Next Steps for the IP QoS Architectures. RFC 2990. IETF (Nov. 2000)

[4] Hewllett-Packard: HP OpenView PolicyXpert. Homepage. Available at: <http://www.openview.hp.com/products/policyexpert/>. Hewllett-Packard Company (2001)

[5] Cisco Systems: Cisco QoS Policy Manager (QPM). Homepage. Available at: <http://www.cisco.com/warp/public/cc/pd/wr2k/qoppmn/>. Cisco Systems (2001)

[6] Extreme Networks: EPICenter 3.0 Technical Specification. Available at: <http://www.extremenetworks.com/products/prod_pdf/EPICenter.pdf>. Extreme Networks (2001)

[7] Saunders, S.: The Policy Makers. Data Communications Magazine (May 1999) 34-56

[8] Clark, R.: The Mechanics of Policy-Based Management. Network Magazine (Mar. 2000) 44-51

[9] Sloman, M.: Policy Driven Management for Distributed Systems. Journal of Network and Systems Management, Vol. 2, Plenum Publishing (Dec. 1994) 333-360

[10] Mahon, H., Bernet, Y., Herzog, S., Schnizlein, J.: Requirements for a Policy Management System. Internet draft <draft-ietf-policy-req-02.txt>. Work in progress. IETF (Nov. 2000)

[11] Hong, J.W.K., Kim, J.S., Park, J.K.: A CORBA-Based Quality of Service Management Framework for Distributed Multimedia Services and Applications. IEEE Network, Vol. 13, No. 2, (1999) 70-79

[12] Granville, L.Z., Fleischmann, R.U., Tarouco, L.M.R., Almeida, M.J.B.: Managing Differentiated Services QoS in End Systems using SNMP. In Proceedings of the 2000 IEEE Workshop on IP-oriented Operations & Management (IPOM 2000). Cracow, Poland (2000) 191-198

[13] Jiang, Y., Tham, C.K., Ko, C.C.: Providing Quality of Service Monitoring: Challenges and Approaches. In Proceedings of the 2000 IEEE/IFIP Network Operations and Management Symposium (NOMS 2000). Honolulu, USA (2000) 115-128

[14] Joshi, R., Tham, C.K.: Integrated Quality of Service and Network Management. In Proceedings of the 2000 IEEE International Conference on Networks (ICON 2000). Singapore (2000) 497

[15] Granville, L.Z., Tarouco, L.M.R.: An Environment to Support QoS Management-Related Tasks on IP Networks. In Proceedings of the 2001 IEEE International Conference on Telecommunications (ICT 2001). Bucharest, Romania (2001)

[16] Distributed Management Task Force. WBEM Initiative. Homepage. Available at: <http://www.dmtf.org/wbem/>.

[17] Martin-Flatin, J.P., Bovet, L., Hubaux, J.P.: JAMAP: a Web-Based Management Platform for IP Networks. In Proceedings of the 10th IFIP/IEEE Internation Workshop on Distributed Systems: Operations & Management (DSOM'99). Zurich, Switzerland. Lecture Notes in Computer Science, Vol. 1700. Springer-Verlag, Berlin Heidelberg New York (1999) 164-178

[18] John, A., Vanderveen, K., Sugla, B.: An XML-based Framework for Dynamic SNMP MIB Extension. In Proceedings of the 10th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'99). Zurich, Switzerland. Lecture Notes in Computer Science, Vol. 1700. Springer-Verlag, Berlin Heidelberg New Your (1999) 107-120

[19] Schoenwaelder, J., Quittek, J., Kappler, C.: Building Distributed Management Applications with the IETF Script MIB. IEEE Journal on Selected Areas in Communications, Special Issue on Network Management and Operations, (2000) Vol. 18, Number 5

Bibliografia

[ALL 2001] ALLOT COMMUNICATIONS. **Managing ISP Point of Presence Traffic**. [S.l.]: Allot communications, Set. 2000. Disponível em: <http://www.allot.com>. Acessado em: 2001.

[APA 2001] APACHE. **Homepage**. Disponível em: <http://www.apache.org>. Acesso em: 07 jan. 2001.

[BAR 2001] BARRETT, D. J.; SILVERMAN, R. **SSH, The Secure Shell: The Definitive Guide**. Sebastopol: O'Reilly & Associates, 2001, 540p.

[BEL 91] BELINA, HOGREFE, SARMA. **SDL with Applications from Protocol Specification.** Munich: Carl Hanser Verlag and Prentice Hall International, 1991.

[BIE 98] BIESZCZAD, A. Mobile Agents for Network Management. **IEEE Communications Surveys**. v.1 n. 1, Fourth Quarter 1998. Disponível em: <http://www.comsoc.org/pubs/surveys>. Acesso em: 27 jul. 2000.

[BLA 98] BLAKE, S.; BLACK, D.; CARLSON, M.; DAVIES, E.; WANG, Z.; WEISS, W. **An Architecture for Differentiated Services**. Request for Comments 2475. IETF (http://www.ietf.org): December 1998.

[BRA 97] BRADEN, R. et al. **Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification**. [S.l.]: IETF, Sept. 1997. (Request for Comments 2205). Disponível em: <http://www.ietf.org>. Acesso em: 6 jan. 1999.

[BRO 97] BROWNLEE, N.; MILLS, C.; RUTH, G. **Traffic Flow Measurement: Architecture**. [S.l.]: IETF, Jan. 1997. (Request for Comments 2063). Disponível em: <http://www.ietf.org>. Acesso em: 23 nov. 2000.

[BRO 97a] BROWNLEE, N. **Traffic Flow Measurement: Meter MIB**. [S.l.]: IETF, Jan. 1997. (Request for Comments 2064). Disponível em: <http://www.ietf.org>. Acesso em: 23 nov. 2000.

[BRO 2001] BROWNLEE, N. **NeTraMet Homepage**. Disponível em: <http://www2.auckland.ac.nz/net//NeTraMet/>. Acesso em: 7 jan. 2001.

[CAS 90] CASE, J. et al. **A Simple Network Management Protocol (SNMP)**. [S.l.]: IETF, May 1990. (STD 15, Request for Comments 1157). Disponível em: <http://www.ietf.org>. Acesso em: 7 jan. 1999.

[CAS 96] CASE, J.; MCCLOGHRIE, K.; ROSE, M.; WALDBUSSER, S. **Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)**. [S.l.]: IETF, Jan 1996. (Request for Comments 1905). Disponível em: <http://www.ietf.org>. Acesso em: nov 1999.

[CIS 2000] CISCO SYSTEMS. **Cisco QoS Policy Manager (QPM) Homepage**. Disponível em: <http://www.cisco.com/warp/public/cc/pd/wr2k/qoppmn>. Acesso em: 22 mar. 2001.

[CIS 2001] CISCO SYSTEMS. **Cisco IOS Command Line Interface (CLI) Tutorial**. Disponível em: <http://www.cisco.com/warp/cpropub/45/tutorial.htm>. Acesso em: 3 mar. 2001.

[CLA 2000] CLARK, R. The Mechanics of Policy-Based Management. **Network Magazine**, p.44-41, Mar. 2000.

[COE 2001] COELHO, G.; GRANVILLE, L. Z.; ALMEIDA, M. J.; TAROUCO, L. Network Executive: uma Ferramenta para Gerenciamento Baseado em Políticas. Em: WORKSHOP DE TMN, WTMN, 6., 2001, Florianópolis. **Anais...** [S.l.:s.n.], 2001.

[COE 2001a] COELHO, G.; GRANVILLE, L. Z.; ALMEIDA, M. J.; TAROUCO, L. M. R. Network Executive: A Policy-Based Network Management Tool. In: IEEE LATIN AMERICAN NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, LANOMS, 2001, Belo Horizonte. **Proceedings…** [S.l.:s.n.], 2001.

[COL 94] COLELLA, R.; CALLON, R.; GARDNER, E.; REKHTER, Y. **Guidelines for OSI NSAP Allocation in the Internet**. [S.l.]: IETF, May 1994. (Request for Comments 1629). Disponível em: <http://www.ietf.org>. Acesso em: maio de 2000.

[COM 91] COMER, Douglas E. **Internetworking with TCP/IP - Principles, Protocols and Architecture**. 2.ed. New Jersey: Prentice-Hall, 1991. v.1.

[CON 2001] CISCO NETWORKS. **Cisco ConfigMaker Documentation**. Disponível em: <http://www.cisco.com/univercd/cc/td/doc/clckstrt/cfgmkr/index.htm>. Acesso em: 3 jan. 2001.

[CRI 2001] WEBTV NETWORKS. **Cricket Homepage**. Disponível em: <http://cricket.sourceforge.net/>. Acesso em: 23 mar. 2001.

[DAM 2001] DAMIANOU, N.; DULAY, N.; LUPU, E.; SLOMAN, M. The Ponder Specification Language. In: WORKSHOP ON POLICIES FOR DISTRIBUTED SYSTEMS AND NETWORKS, POLICY, 2001. **Proceedings…** [S.l.; s.n.], 2001.

[DER 2001] DERI, L.; CARBONE, R.; SUIN, S. Monitoring Networks Using ntop. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, IM, 2001, Seattle-USA. **Proceedings…** [S.l.;s.n.], 2001.

[DIS 2001] DISTRIBUTED MANAGEMENT TASK FORCE. **WBEM Initiative Homepage**. Disponível em: <http://www.dmtf.org/wbem/>. Acesso em: jul. 2001.

[DUR 2000] DURHAM, D. et al. **The COPS (Common Open Policy Service) Protocol**. [S.l.]: IETF, Jan. 2000. (Request for Comments 2748). Disponível em: <http://www.ietf.org>. Acesso em: 2000.

[DUT 95] DUTTON, Harry J. R.; LENHARD, Petter. **Asynchronous Transfer Mode (ATM): Technical Overview**. 2.ed. New Jersey: Prentice Hall, 1995.

[EDE 2001] EDER, M.; NAG, S. **Service Management Architectures Issues and Review**. [S.l.]: IETF, Jan. 2001. (Request for Comments 3052). Disponível em: <http://www.ietf.org>. Acesso em: 19 fev. 2001.

[ETS 98] ETSI. **General Packet Radio Service (GPRS):** Service Description, Stage 2, v.6.1.1. [S. l.], 1998. (GSM03.60).

[EXT 2001] EXTREME NETWORKS. **EPICenter 3.0 Technical Specification**. Disponível em: <http://www.extremenetworks.com/products/prod pdf/EPICenter.pdf>. Acesso em: 21 mar. 2001.

[FEN 97] FENNER, W. **Internet Group Management Protocol, Version 2**. [S.l.]: IETF, Nov. 1997. (Request for Comments 2236). Disponível em: <http://www.ietf.org>. Acesso em: 14 dez. 1999.

[FLA 99] MARTIN-FLATIN, J. P. Push vs. pull in Web-based network management. In: SIXTH IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, IM, 1999. **Proceedings…** [S.l.:s.n.], 1999.

[GOL 96] GOLDSZMIDT, G. **Distributed Management by Delegation**. Columbia University, United States, 175 pages, april 1996. PhD thesis.

[GRA 99] GRANVILLE, L. Z.; ULBRICH, L.; TAROUCO, L. Gerenciamento de Redes de Alta Velocidade. In: TELEMÁTICA, 1999. **Anais...** Porto Alegre:[s.n.], 1999.

[GRA 2000] GRANVILLE, Lisandro Z. **Gerenciamento de QoS em Redes de Alta Velocidade**. Exame de Qualificação. Porto Alegre: PPGC da UFRGS, 2000. 90p.

[GRA 2000a] GRANVILLE, L. Z.; TAROUCO, L. High-Speed Networks QoS Management: Where to Put the Complexity? In: INET, 2000, Yokohama-Japan. **Proceedings…**[S.l.:s.n.], 2000.

[GRA 2000b] GRANVILLE, L. Z. et al. Managing Differentiated Services QoS in End Systems using SNMP. In: IEEE WORKSHOP ON IP-ORIENTED OPERATIONS & MANAGEMENT, IPOM, 2000, Cracrow-Poland. **Proceedings…** New York:IEEE Press, 2000.

[GRA 2000c] GRANVILLE, L. Z. et al. Management of Differentiated Services through QoS-enabled Hosts. In: ASIA-PACIFIC NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, APNOMS, 4., 2000, Nara-Japan. **Proceedings…** NewYork:IEEE Press, 2000.

[GRA 2000d] GRANVILLE, L. Z. et al. Management of Networks with End-to-End Differentiated Service QoS Capabilities. In: BIENNIAL INTERNATIONAL CONFERENCE ON ADVANCES IN INFORMATION SYSTEMS, ADVIS, 1., 2000, Izmir-Turkey. **Proceedings…** Berlin:Spring Verlarg, 2000.

[GRA 2000e] GRANVILLE, L. Z. et al. NetPlus – Uma Ferramenta para Gerenciamento de QoS via Web. In: WORKSHOP DE PORTO ALEGRE SOBRE INTERNET2 E REDES DE ALTA VELOCIDADE, 1., 2000, Porto Alegre. **Anais...** [S.l.:s.n.], 2000.

[GRA 2000f] GRANVILLE, L. Z.; ROCHOL, J.; TAROUCO, L. M. R. Arquitetura para gerenciamento de redes ATM através de SNMP. In: ARGENTINE SYMPOSIUM ON COMPUTING TECHNOLOGY, AST, 2000. **Proceedings…** [S.l.:s.n.], 2000.

[GRA 2001] GRANVILLE, L. Z.; TAROUCO, L. Management of QoS Provisioning Architectures. In: INTERNATIONAL CONFERENCE ON TELECOMMUNICATION SYSTEMS, 9., 2001, Dallas. **Proceedings…** [S.l.:s.n.], 2001.

[GRA 2001a] GRANVILLE, L. Z.; TAROUCO, L. QAME - An Environment to Support QoS Management Related Tasks on IP Networks. In: INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS, ICT, 2001, Bucharest-Romania. **Proceedings…** [S.l.:s.n.], 2001.

[GRA 2001b] GRANVILLE, L. Z. et al. Integrated Management of QoS-enabled Networks using QAME. In: INTERNATIONAL CONFERENCE ON NETWORKING, ICN, 2001, Colmar-France. **Proceedings…** [S.l.:s.n.], 2001.

[GRA 2001c] GRANVILLE, L. Z. et al. NetPlus - Um ambiente para gerência de QoS baseado na Web. Em: WORKSHOP RNP2, WRNP2, 3., 2001, Florianópolis. **Anais…** [S.l.:s.n.], 2001.

[GRA 2001d] GRANVILLE, L. Z. et al. NetPlus - Um ambiente para gerência de QoS baseado na Web. Em: NewsGeneration. [S.l.:s.n.]. Disponível em: <http://www.rnp.br/newsgen>. Acesso em: 3 ago. 2001.

[GRA 2001e] GRANVILLE, L. Z.; TAROUCO, L. M. R. QAME: QoS-Aware Management Environment In: INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, COMPSAC, 2001, Chicago. **Proceedings…** [S.l.:s.n.], 2001.

[GRA 2001f] GRANVILLE, L. Z. et al. QoS Management-Related Tasks: Beyond Policy-Based Management. In: IEEE LATIN AMERICAN NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, LANOMS, 2001, Belo Horizonte. **Proceedings…** [S.l.:s.n.], 2001.

[GRA 2001g] GRANVILLE, L. Z. et al. An Approach for Integrated Management of Networks with Quality of Service Support Using QAME. In: IFIP/IEEE INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS & MANAGEMENT, DSOM, 2001, Nancy-France. **Proceedings…** [S.l.:s.n.], 2001.

[HAH 2000] HAHN, S.; STEVENS, M. **Resource Allocation Protocol (rap) IETF Working Group**. Disponível em: <http://www.ietf.org/html.charters/rap-charter.html>. Acesso em: 2000.

[HAL 2000] HALPERN, J.; ELLESSON, E. **Policy Framework (policy) IETF Working Group**. Disponível em: <http://www.ietf.org/html.charters/policy-charter.html>. Acesso em: 2000.

[HEI 99] HEINANEN, J.; BAKER, F.; WEISS, W.; WROCLAWSKI, J. **Assured Forwarding PHB Group**. [S.l.]: IETF, Jun. 1999. (Request for Comments 2597). Disponível em: <http://www.ietf.org>. Acesso em: abril 2000.

[HER 2000] HERZOG, S. **RSVP Extensions for Policy Control**. [S.l.]: IETF, Jan. 2000. (Request for Comments 2750). Disponível em: <http://www.ietf.org>. Acesso em: 2000.

[HEW 2001] HEWLETT-PACKARD. **HP OpenView Homepage**. Disponível em: <http://openview.hp.com>. Acesso em: 19 mar. 2001.

[HEW 2001a] HEWLETT-PACKARD. **HP OpenView PolicyXpert Homepage**. Disponível em: <http://www.openview.hp.com/products/policyexpert/>. Acesso em: 20 mar. 2001.

[HON 99] HONG, J. W. K.; KIM, J. S.; PARK, J. K. A CORBA-Based Quality of Service Management Framework for Distributed Multimedia Services and Applications. **IEEE Network**, v.13, n.2, p.70-79, 1999.

[HUS 2000] HUSTON, G. **Next Steps for the IP QoS Architecture**. [S.l.]: IETF Nov. 2000. (Request for Comments 2990). Disponível em: <http://www.ietf.org>. Acesso em: 27 fev. 2001.

[HUT 94] HUTCHISON, D.; COULSON, G.; CAMPBELL, A.; BLAIR, G. Quality of Service Management in Distributed Systems. In: **Network and Distributed Systems Management**. M. Sloman ed.: Addison Wesley, 1994. chap.11.

[IET 2001] INTERNET ENGINEERING TASK FORCE. **IETF Homepage**. URL: http://www.ietf.org. june 2000.

[INT 2001] INTERNET2 PROJECT. **Homepage**. Disponível em: <http://www.internet2.org>. Acesso em 7 jan. 2001.

[ITU 98] ITU-T. **Packet-based Multimedia Communications Systems.** Geneve: ITU-T, Recommendation H.323, Jan. 1998.

[JAC 99] JACOBSON, V.; NICHOLS, K.; PODURI, K. **An Expedited Forwarding PHB**. [S.l.]: IETF, Jun. 1999. (Request for Comments 2598). Disponível em: <http://www.ietf.org>. Acesso em: maio 2000.

[JIA 2000] JIANG, Y.; THAM, C. K.; KO, C. C. Providing Quality of Service Monitoring: Challenges and Approaches. In: IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SEMINAR, NOMS, 2000, USA. **Proceedings…** [S.l.:s.n.], 2000.

[JOH 99] JOHN, A.; VANDERVEEN, K.; SUGLA, B. An XML-based Framework for Dynamic SNMP MIB Extension. In: IFIP/IEEE INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS & MANAGEMENT, DSOM, 1999, Zurich-Switzerland. **Proceedings…** Lecture Notes in Computer Science, Vol. 1700. Springer-Verlag, Berlin Heidelberg New Your (1999) 107-120

[JOS 2000] JOSHI, R.; THAM, C.K. Integrated Quality of Service and Network Management. In: IEEE INTERNATIONAL CONFERENCE ON NETWORKS, ICON, 2000, Singapore. **Proceedings…**[S.l.;s.n], 2000.

[LEI 96] LEINWAND, Allan, CONROY, Karen F. **Network Management: A Practical Perspective**. 2.ed. Harlow: Addison-Wesley, 1996.

[LIN 91] LIN, A.; SILVESTER, J. Priority Queuing Strategies and Buffer Allocation Protocols in Traffic Control at an ATM Integrated Broadband Switching System. **IEEE JSAC**, v.9, n.9, Dec. 1991.

[LON 2000] LONEY, K.; KOCH, G. **Oracle8I : The Complete Reference**. 10.ed. Oracle Press, May 2000. 1313p.

[LUP 99] LUPU, E.; SLOMAN, M. Conflicts in Policy-based Distributed Systems Management. **IEEE Transactions on Software Engineering**, special issue on inconsistency management, v.25, n.6, p.852-869, Nov. 1999.

[MAC 2001] MACFADEN, M.; SAPERIA, J.; TACKABURY, W. **Configuring Networks and Devices With SNMP**. [S.l.]: IETF, May 2001. (Internet draft <draft-ietf-snmpconf-bcp-05.txt> work in progress). Disponível em: <http://www.ietf.org>. Acesso em: 2 jun. 2001.

[MAC 2001a] MACROMEDIA FLASH 5. **Homepage**. Disponível em: <http://www.macromedia.com/software/flash/>. Acesso em: 30 jan. 2001.

[MAH 2000] MAHON, H.; BERNET, Y.; HERZOG, S.; SCHNIZLEIN, J. **Requirements for a Policy Management System**. [S.l.]: IETF, Nov. 2000. (Internet draft <draft-ietf-policy-req-02.txt> work in progress). Disponível em: <http://www.ietf.org>. Acesso em: 19 jan. 2001.

[MAR 99] MARTIN-FLATIN, J.P.; BOVET, L.; HUBAUX, J.P. JAMAP: a Web-Based Management Platform for IP Networks. In: IFIP/IEEE INTERNATION WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS & MANAGEMENT, DSOM, 1999, Zurich-Switzerland. **Proceeding…** Lecture Notes in Computer Science, Vol. 1700. Springer-Verlag, Berlin Heidelberg New York (1999) 164-178.

[MCB 96] McBRIDE, D. **The SLA Cookbook**: A Recipe for Understanding System and Network Resource Demands. [S.l.]:Hewlett-Packard Company, 1996. Disponível em: <http://www.hp.com/openview/ rpm>. Acesso em: 1999.

[MCC 90] MCCLOGHRIE, K.; ROSE, M. **Management Information Base for Network Management of TCP/IP-based internets**. [S.l.]: IETF, May 1990. Disponível em: <http://www.ietf.org>. Acesso em: abril, 1998.

[MCD 99] McDYSAN, David. **Qos and Traffic Management in IP and ATM Networks**. NewYork: McGraw Hill, 1999. 456p.

[MIN 98] MINOLI, D.; MINOLI, E. **Delivering Voice over IP Networks**. New York: Wiley Computer Publishing, 1998.

[MIT 2000] MITZEL, D. **Overview of 2000 IAB Wireless Internetworking Workshop**. [S.l.]: IETF, 2000. (Request for Comments 3002). Disponível em <http://www.ietf.org>. Acesso em: 2000.

[MOF 93] MOFFETT, J.; SLOMAN, M. Policy Hierarchies for Distributed Systems Management. **IEEE Journal on Selected Areas in Communications**, v.11, n.9, p.1404-1414, Dec. 1993.

[NIC 98] NICHOLS, K.; BLAKE, S.; BAKER, F.; BLACK, D. **Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers**. [S.l.]: IETF, Dec. 1998. (Request for Comments 2474). Disponível em: <http://www.ietf.org>. Acesso em: 1999.

[NIC 99] NICHOLS, K.; JACOBSON, V.; ZHANG, L. **A Two-bit Differentiated Services Architecture for the Internet**. [S.l.]: IETF Nov. 1999. (Request for Comments 2638). Disponível em: <http://www.ietf.org>. Acesso em: 2000.

[NS 2001] NS NETWORK SIMULATOR. Disponível em: <http://www.isi.edu/nsnam/ns>. Acesso em: 20001.

[OET 2001] OETIKER, T.; RAND, D. **MRTG – Multi Router Traffic Grapher**. Disponível em: <http://www.mrtg.org>. Acesso em: 2001.

[PAN 96] PANTTAJA, J.; PANTTAJA, M.; BOWMAN, J.; BOWMAN, J. S. **The Sybase SQL Server Survival Guide**. New York: John Wiley & Sons, May 1996. 384p.

[PHP 2001] PHP HYPERTEXT PREPROCESSOR. **Homepage**. Disponível em: <http://www.php.net>. Acesso em: 2001.

[PIN 2000] PINNES, E. L. Operations and Management for Next Generation Networks (tutorial). In: ASIA-PACIFIC NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, APNOMS, 2000, Nara-Japan. **Proceedings…** New York:IEEE Press, 2000. Tutorial.

[QUA 2000] QUALCOMM. **Homepage**. Disponível em <http://www.qualcomm. com>. Acesso em: 2000.

[QUA 2000a] QUADROS, G.; ALVES, A.; MONTEIRO, E.; BOAVIDA, F. How Unfair can Weighted Fair Queuing be? In: FIFTH IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS, ISCC, 2000, Antibes-France. **Proceedings…** [S.l.:s.n.], 2000.

[QUA 2001] THE QUALCOMM High Data Rate Wireless Network. Disponível em <http://www.qualcomm.com/hdr/>. Acesso em: 2001.

[QOS 99] QOS FORUM. **Introduction to QoS Policies**. White Paper. Disponível em: <http://www.qosforum.com>. Acesso em: 1999.

[QOS 2001] QOS FORUM. **Homepage**. Disponível em: <http://www.qosforum.com>. Acesso em: 2001.

[QUI 99] QUITTEK, J.; KAPPLER, C. Remote Service Deployment on Programmable Switches with the IETF SNMP Script MIB. In: IFIP/IEEE INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS AND MANAGEMENT, DSOM, 2000. **Proceedings…** Berlin:Springer Verlag, 1999.

[RIB 2001] RIBEIRO, M. B.; GRANVILLE, L. Z.; TAROUCO, L. M. R.; ALMEIDA, M. J. B. Distributed Monitoring Over IP Networks. In: WORLD MULTICONFERENCE ON SYSTEMICS, CYBERNETICS AND INFORMATICS, SCI, 5., 2001, Orlando . **Proceedings…** [S.l.:s.n.], 2001.

[RIB 2001a] RIBEIRO, M. B.; GRANVILLE, L. Z.; ALMEIDA, M. J. B.; TAROUCO, L. M. R. QoS Monitoring System on IP Networks. In: IFIP/IEEE INTERNATIONAL CONFERENCE ON MANAGEMENT OF MULTIMEDIA NETWORKS AND SERVICES, MMNS, 2001, Chicago. **Proceedings…** [S.l.:s.n.], 2001.

[ROS 2001] ROSEN, E. C.; VISWANATHAN, A.; CALLON, R. **Multiprotocol Label Switching Architecture**. [S.l.]: IETF, Jan. 2001. (Request for Comments 3031). Disponível em: <http://www.ietf.org>. Acesso em: 28 fev. 2001.

[SAP 2000] SAPERIA, J.; PARTAIN, D. **Configuration Management with SNMP (snmpconf) IETF Working Group**. Disponível em: <http://www.ietf.org/html.charters/snmpconf-charter.html>. Acesso em: 2000.

[SAU 99] SAUNDERS, S. The Policy Makers. **Data Communications Magazine**, p.34-56, May 1999.

[SCH 96] SCHULZRINNE, H. et al. **RTP: A Transport Protocol for Real-Time Applications**. [S.l.]: IETF, Jan. 1996. (Request for Comments 1889). Disponível em: <http://www.ietf.org>. Acesso em: 1998.

[SHE 97] SHENKER, S.; WROCLAWSKI, J. **General Characterization Parameters for Integrated Service Network Elements**. [S.l.]: IETF, Sept. 1997. (Request for Comments 2215). Disponível em: <http://www.ietf.org>. Acesso em: 1999.

[SIA 98] SIAMWALLA, R.; SHARMA, R.; KESHAV, S. **Discovering Internet Topology.** (tech. report, unpublished manuscript, Cornell University). Disponível em: <http://www.cs.cornell.edu/skeshav/>. Acesso em: 27 out. 2000.

[SLO 94] SLOMAN, M. Policy Driven Management For Distributed Systems. **Journal of Network and Systems Management**, v.2, n.4, p.333-360, Dec. 1994.

[SNI 2000] SNIR, Y. et al. **Policy Framework QoS Information Model**. [S.l.]: IETF, Nov. 2000. (Internet draft <draft-ietf-policy-qos-info-model-02.txt> work in progress). Disponível em: <http://www.ietf.org>. Acesso em: 19 jan. 2001.

[STA 96] STALLINGS, W. **RMON2 – The Next Generation of Remote Network Monitoring.** [S.l.]:ConneXions, 1996.

[STA 98] STALLINGS, W. **ISDN and Broadband ISDN With Frame Relay and ATM**. 4.ed. New Jersey: Prentice Hall, 1998.

[STA 99] STALLINGS, W. **SNMP, SNMPv2, SNMPv3 and RMON 1 and 2**. 3.ed. Harlow: Addison-Wesley, 1999.

[STR 2001] STRASSNER, J.; WESTERINEN, A.; ELLESSON, E.; MOORE, B.; MOATS, R. **Policy Core LDAP Schema**. [S.l.]: IETF, May 2001. (Internet draft <draft-ietf-policy-core-schema-11.txt> work in progress). Disponível em: <http://www.ietf.org>. Acesso em: 03 jun. 2001.

[TAN 96] TANENBAUM, Andrew S. **Computer Networks**. 3.ed. New Jersey: Prentice Hall, 1996. 814p.

[TEL 2001] TELCHEMY. **Monitoring Voice Quality in Voice Over IP Networks**. Disponível em: <http://netgroup-serv.polito.it/winpcap/>. Acesso em: 2001.

[TSY 2001] TSYKIN, M. On Web Quality of Service: Approaches to Measurement of End-to-End Response Time. In: INTERNATIONAL CONFERENCE ON NETWORKING, ICN, 2001, Colmar-France. **Proceedings…** [S.l.:s.n.], 2001.

[TU 99] TU BRAUNSCHWEIG, NEC C&C RESEARCH LABORATORIES. **Jasmin - A Script MIB Implementation**. Disponível em: <http://www.ibr.cs.tubs. de/projects/jasmin/>. Acesso em: 1999.

[UNI 96] UNIVERSITY OF MICHIGAN. **The SLAPD and SLURPD Administrator's Guide.** LDAP Development Team, 1996. Disponível em: < http://www.openldap.org>.

[UNI 2001] UNIVERSITY OF CALIFORNIA DAVIS. **NET-SNMP project home page**. Disponível em: <http:// net-snmp.sourceforge.net>. Acesso em: 2001.

[VIC 98] **VIC: User Guide for v2.8**. London:University College London, Computer Science Department. Disponível em: <http://www.ice.cs.ucl. ac.uk/multimedia/software/>. Acesso em: 1998.

[WAL 97] WALDBUSSER, S. **Remote Network Monitoring Management Information Base Version 2 using SMIv2**. [S.l.]: IETF, Jan. 1997. (Request for Comments 2021). Disponível em: <http://www.ietf.org>. Acesso em: 11 jul. 2000.

[WAL 2000] WALDBUSSER, S. **Remote Network Monitoring Management Information Base**. [S.l.]: IETF, May 2000. (STD 59, Request for Comments 2819). Disponível em: <http://www.ietf.org>. Acesso em: 11 jul. 2000.

[WEL 95] WELSH, M.; KAUFMAN, L. **Running LINUX**. Sebastopol, CA: O'Reilly & Associates, 1995.

[WIN 2000] WINSOR, J.; FREEMAN, B. **Jumping JavaScript**. [S.l.]:Sun Microsystems Press, 2000.

[WRO 97] WROCLAWSKI, J. **The Use of RSVP with IETF Integrated Services**. [S.l.]: IETF, Sept. 1997. (Request for Comments 2210). Disponível em: < http://www.ietf.org>. Acesso em: 9 set. 2000.

[XU 2001] XU, D.; NAHRSTEDT, K.; WICHADAKUL, D. QoS-Aware Discovery of Wide-Area Distributed Services. In: IEEE/ACM INTERNATIONAL SYMPOSIUM ON CLUSTER COMPUTING AND THE GRID, CCGrid, 2001, Brisbane-Australia. **Proceedings…** [S.l.:s.n.], 2001.

[YAR 99] YARGER, R. J.; REESE, G.; KING, T. **MySQL & mSQL**. [S.l.]:O'Reilly, 1999.

[YAV 99] YAVTKAR, R.; HOFFMAN, D.; BERNET, Y.; BAKER, F. **SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks**. [S.l.]: IETF, May 1999. (Internet draft <draft-ietf-issll-is802-sbm-08.txt> work in progress). Disponível em <http://www.ietf.org). Acesso em: 2000.

[YAV 2000] YAVATKAR, R.; PENDARAKIS, D.; GUERIN, R. **A Framework for Policy Based Admission Control**. [S.l.]: IETF, Jan. 2000. (Request for Comments 2753). Disponível em: <http://www.ietf.org>. Acesso em: 29 out. 2000.

[YUN 97] YUN, T. H.; KONG, J. Y.; HONG, J. W. A CORBA-based Distributed Multimedia System, In: PACIFIC WHOKSHOP ON DISTRIBUTED MULTIMEDIA SYSTEMS, Vancouver-Canada. **Proceedings…** [S.l.:s.n.], 1997.