Grupo de Trabalho em Configurações de Redes

Diagnóstico e Alternativas

Lisandro Zambenedetti Granville

Clarissa Marquezan

Ricardo Vianna

Rodrigo Sanger

Augusto Bueno

Douglas Nascimento

Novembro de 2003

Documento de Diagnóstico e Alternativas do

Grupo de Trabalho em Configurações de Rede (GT-Config) da RNP2

Índice

Diagnóstico e Alternativas 1
Índice 3
Lista de Figuras 4
1 Introdução 5
2 Gerenciamento Tradicional e Definição do Programa 6
2.1 Gerenciamento orientando a dispositivos 6
2.2 Gerenciamento de Redes Baseado em Políticas (PBNM) 7
2.3 PBNM, pesquisas e o mercado de soluções de gerência 9
2.4 Análise sobre a abrangência do PBNM 10
2.5 Definição do Problema 10
3 QAME (QoS-Aware Management Environment) 12
3.1 Ambiente baseado na Web 12
3.2 Políticas no contexto do IETF 13
3.3 Suporte Preliminar às Políticas do IETF 14
4 Web Services 15
4.1 A pilha de tecnologia para Web Services 15
4.1.1 Rede 16
4.1.2 Transporte 16
4.1.3 Mensagem 16
4.1.4 Descrição 16
4.1.5 Descoberta 17
4.2 Segurança dos Web Services 17
4.3 Web Services e PHP 17
5 Hierarquia de configurações de redes 20
6 Alternativas 22
6.1 Aplicação de gerenciamento 22
6.2 PDPs 22
6.3 Representação de políticas 23
6.4 Armazenamento de políticas 23
7 Conclusões 24
8 Referências 25

Lista de Figuras

Figure 1. Exemplo de uma solução de gerência orientada a dispositivos 7
Figure 2. Arquitetura para gerenciamento de redes baseado em políticas..... 8
Figure 3. PDP, PEP e estação de gerenciamento..... 9
Figure 4. Interface Gráfica do QAME..... 13
Figure 5. Hierarquia de Tecnologias Web Services..... 16
Figure 6. Cliente Web Service 18
Figure 7. Servidor Web Service..... 19
Figure 8. Resultado da consulta ao Web Service..... 19
Figure 9. Proposta de arquitetura para gerência de configuração na RNP 20

1 Introdução

O Grupo de Trabalho em Configuração de Redes (GT-Config) da RNP foi criado com o objetivo principal de a investigar, implementar, testar e validar um sistema piloto para a automação de configuração de dispositivos de rede, prioritariamente em relação a QoS, mas com vistas à configuração de multicast e segurança. Assim, o GT-Config procura facilitar o processo de configuração dos dispositivos do backbone da RNP, através da automatização da configuração de equipamentos.

A automação das configurações é baseada no paradigma de gerenciamento de redes baseado em políticas, onde administradores de redes definem políticas de operação em alto nível, e o sistema de gerência providencia a configuração dos dispositivos de forma a alcançar os comportamentos expressos nas políticas. O produto final será um sistema hierárquico capaz de configurar dispositivos de acordo com políticas de rede, e com isso obter-se a abstração das particularidades de acesso aos equipamentos que faz com que, atualmente, as configurações sejam executadas quase sempre de forma manual, e por isso, lenta.

As atividades do GT-Config estão sendo realizadas de forma articulada e em parceria com as atividades dos seguintes projetos:

- **IQoM** (Infra-estrutura para Medição de QoS e Implantação de Serviços Diferenciados), cujos integrantes são: UNIFACS, UFRGS, UFSC, UFPR, Fundação CPqD, Universidade de Cambrigde (Intel Research) e FURG;
- **MetroPOA-II** (Rede Metropolitana da Grande Porto Alegre - Fase II), cujos integrantes são: UFRGS, PUC-RS, UNISINOS e Brasil Telecom;
- **SCQoS** (Sistema para Configuração de QoS em Redes IP), cujos integrantes são: Fundação CPqD e UFRGS.

Além destas parcerias, o GT-Config possui relação direta com os anteriores **GT-QoS** e **GT-Diretórios**, e com os atuais GTs Qualidade de Serviços 2 (**GT-QoS2**), Vídeo Digital 2 (**GTVD**), Voz sobre IP Avançado (**GT-VoIP**) e Diretório para Educação (**DIR-EDU**).

Neste relatório são apresentados os resultados do estudo das tecnologias para configuração de dispositivos e da arquitetura para gerenciamento baseado em políticas. Da mesma forma, são apresentadas as alternativas para a criação do piloto, que envolve o desenvolvimento de um sistema de configuração, bem como sua avaliação em um ambiente de produção real.

2 Gerenciamento Tradicional e Definição do Programa

O gerenciamento de redes tradicional lida com um conjunto de informações 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 o gerenciamento mais complexo. Além disso, o volume de informações é proporcional ao tamanho das redes, 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 gerenciamento 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 gerenciamento extremamente grande e complexo, e que não pode mais ser tratado pelas soluções de gerenciamento de redes tradicionais [EDE 2001].

Nesta seção serão revistos o modelo de gerenciamento tradicional orientado a dispositivos, o gerenciamento de redes baseado em políticas, e uma revisão sobre as soluções de gerenciamento de QoS. Por fim, a seção encerra com a definição do principal problema investigado pelo GT-Config: a falta de automatização do processo de configuração de dispositivos em redes com QoS.

2.1 Gerenciamento orientando a dispositivos

As plataformas de gerenciamento 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 Figure 1 apresenta a interface gráfica da plataforma de gerência HP OpenView [HEW 2003] 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 a rede gerenciada, o administrador acaba sendo muitas vezes 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 gerenciamento pode ser utilizado, mas em redes maiores seria impossível a um administrador analisar todas as informações existentes.

Nesse contexto, o gerenciamento orientado a dispositivos, ainda que precioso 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.

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 configuração, complementado o gerenciamento orientado a dispositivos das plataformas atuais, através de soluções que forneçam uma visão abrangente da rede.

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

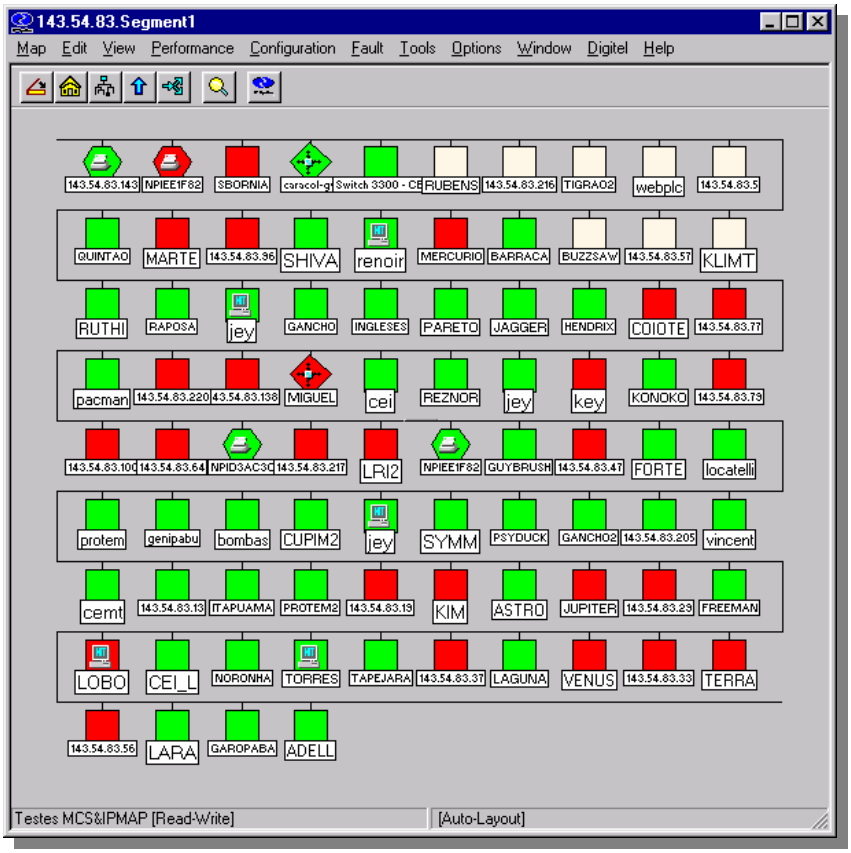

Figure 1. Exemplo de uma solução de gerência orientada a dispositivos

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 gerenciamento a serem executadas, principalmente em relação à configuração de dispositivos. Quando o administrador da rede passa a lidar com muitas estruturas diferentes, o gerenciamento passa a ser complexo, e conseqüentemente mais difícil de ser mantido.

O uso de PBNM pode ser comparado ao uso das linguagens de programação. Nesse 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 gerenciamento a serem usadas. O sistema de gerenciamento preocupa-se em interpretar estas políticas e implantá-las na rede. O suporte a QoS, multicast e segurança em redes heterogêneas é rico em diversidade de tecnologias. A gerência de todas as tecnologias necessárias é complexa, e o uso de políticas para o gerenciamento desses serviços, 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 gerenciamento, porque permite que políticas complexas possam ser formadas pela combinação de várias políticas simples.

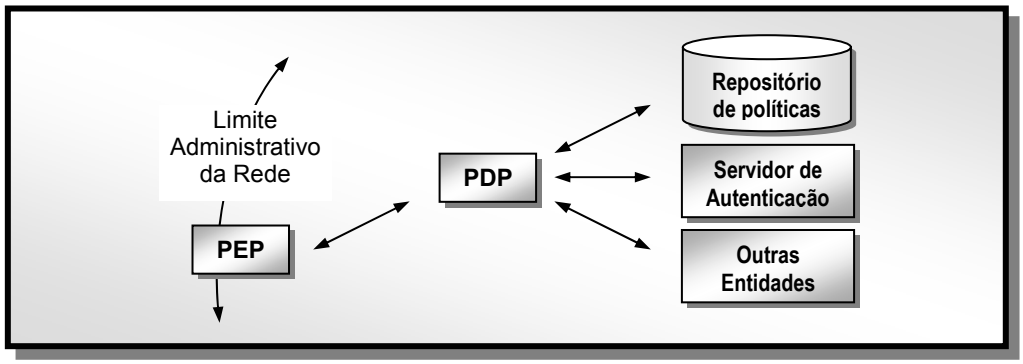

Figure 2. Arquitetura para gerenciamento de redes baseado em políticas

Como definido pelo grupo de trabalho policy (Policy Framework) [HAL 2003] do IETF, uma arquitetura para gerenciamento baseada em políticas 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 (Figure 2).

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 Figure 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.

O gerenciamento de configuração se dá através da interação entre os PDPs e o ambiente de gerenciamento da rede. Uma estação de gerenciamento 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 gerenciamento 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 gerenciamento acessa a PIB de cada PDP através de um protocolo de gerenciamento (e.g. SNMP). A visão do gerente é obtida, por exemplo, 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 (Figure 3).

Uma característica importante do gerenciamento baseado em políticas é que ele pode ser aplicado em conjunto com o gerenciamento 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 PDP associado para que tratem adequadamente o fluxo de vídeo. Esta programação pode ser realizada através da comunicação, via

SNMP, do PDP 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.

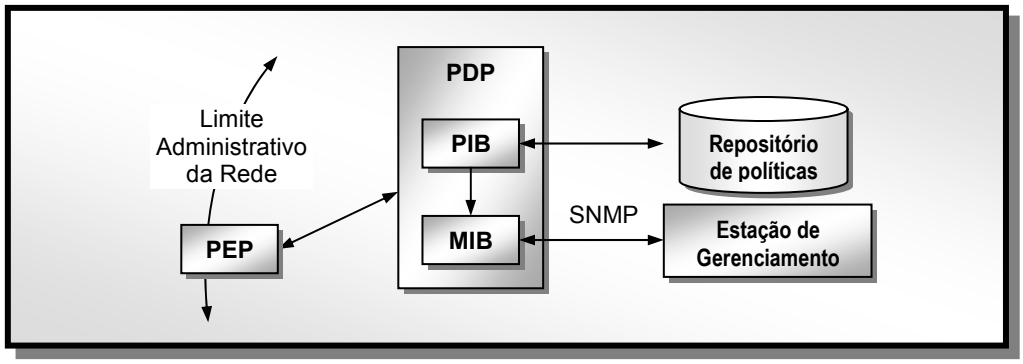

Figure 3. PDP, PEP e estação de gerenciamento

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 o pioneiro 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 deste GT-Config, 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. Esse trabalho do IETF é uma das bases do modelo de gerenciamento proposto no GT, 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 2003a] 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.

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 2002] 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, enquanto que o Orchestream Enterprise da Orchestream Networks é baseado em Oracle. 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 indicadas 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. Outras atividades relacionadas à gerência de QoS devem ser realizadas, ainda que o PBNM esteja presente na rede gerenciada, como por exemplo a medição, que é alvo de estudo do GT-QoS.

2.5 Definição do Problema

O GT-Config tem como principal objetivo propor uma solução para a automação de configurações em redes maiores e com QoS, como a rede da RNP. A automação de configuração pode ser alcançada de várias formas (e.g. transferência de scripts, agentes móveis, código ativo, etc.). No GT-Config propõe-se o uso do gerenciamento de redes baseado em políticas como mecanismo de automação das configurações.

O PBNM define que políticas de rede (expressas através de uma linguagem com alto grau de abstração) são utilizadas para que os administradores expressem os objetivos e metas da rede gerenciada. Tais políticas são armazenadas em um repositório de políticas (implementado, por exemplo, em um serviço de diretórios baseado em LDAP) e transferidas para o sistema de gerenciamento PBNM para serem aplicadas na rede. O sistema PBNM traduz, em momentos pré-definidos, as políticas descritas em alto nível em ações de configuração dos dispositivos, utilizando os diversos protocolos de configuração existentes na rede (e.g. Telnet/CLI, SNMP ou HTTP). Nesse contexto, o administrador de rede expressa nas políticas os objetivos e metas, mas é o sistema PBNM que se encarrega de efetivamente configurar os dispositivos de forma automatizada, liberando os administradores de uma intervenção manual.

A arquitetura PBNM mais comumente aceita atualmente considera que as políticas de rede atuam dentro de um único domínio administrativo, que possui uma única lógica de gerência e um ponto central de decisões. No caso da RNP, entretanto, vários pontos de decisão são encontrados ao longo dos vários POPs. Um operador de POP, por exemplo, pode não preferir uma política de rede que privilegie uma transmissão de vídeo, de forma a permitir uma melhor navegação Web por parte de seus usuários. Neste contexto, o PBNM “clássico” precisa ser revisto e adaptado para o ambiente de gerenciamento apresentado pela RNP.

3 QAME (QoS-Aware Management Environment)

O QAME (QoS-Aware Management Environment) é uma ambiente para gerenciamento de redes que leva em conta, explicitamente, os aspectos relacionados com QoS. Nessa seção são apresentadas as características do QAME que mais podem contribuir para a futura arquitetura do piloto a ser realizado.

3.1 Ambiente baseado na Web

O QAME é um ambiente de gerenciamento de redes baseado na Web e que resultou da evolução do sistema NetPlus [GRA 2001], também baseado na Web.

Elementos de rede necessitam ser administrados, monitorados, analisados e avaliados. Essas atividades são necessárias para que a ligação entre a rede e seus usuários mantenha-se sempre ativa e funcionando da melhor forma possível. Abaixo são citados alguns exemplos de atividades que podem ser realizadas através do QAME.

- O gerente pode visualizar os dispositivos da rede através de um mapa topológico.
- O gerente pode acessar, consultar e configurar os dispositivos de rede.
- O gerente pode obter gráficos diários, semanais ou anuais contendo estatísticas sobre o volume de tráfego nos dispositivos.

O servidor Web é responsável por receber requisições HTTP do cliente (navegador web) e proceder com consultas ao banco de dados ou com consultas diretas aos dispositivos gerenciados. A consulta ao banco de dados é realizada utilizando-se scripts PHP4. O banco escolhido foi o MySQL por ser rápido e não consumir muitos recursos de máquina, se comparado aos bancos tradicionais.

O navegador Web utilizado deve possuir suporte à tecnologia Flash versão 5.0. 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.

A Figure 4 apresenta a interface gráfica baseada na Web do QAME, mostrando a topologia principal de uma rede de testes. 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 alterar a disposição gráfica dos elementos e remover elementos e ligações. Por fim, no modo ligação é possível criar ligações entre os elementos de uma topologia.

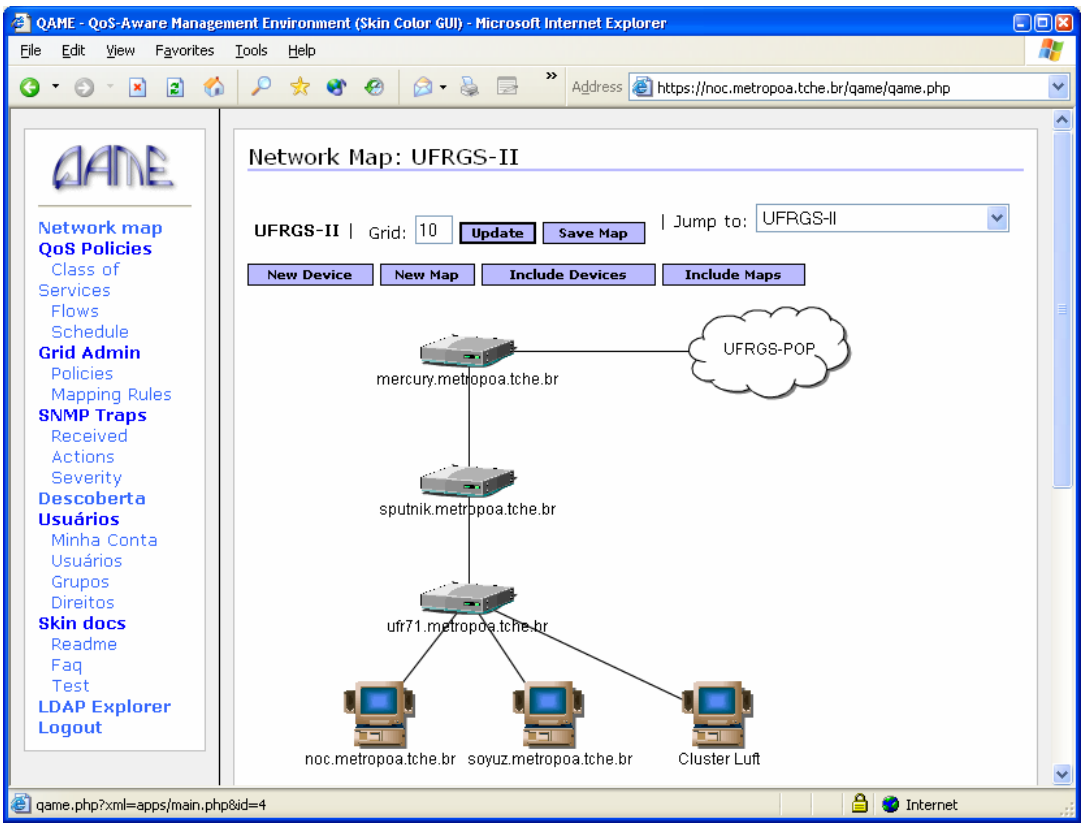

Figure 4. Interface Gráfica do QAME

3.2 Políticas no contexto do IETF

Os trabalhos do IETF, atualmente, estão mais voltados para a representação de políticas do que com o sistema de gerenciamento baseado em políticas. Políticas do IETF são representadas através do armazenamento de dados em servidores LDAP com esquemas para políticas. O primeiro e mais importante esquema LDAP para suporte a políticas é o PCIM (*Policy-Core Information Model*) [MOR 2001]. Esse esquema é um a extensão do CIM (*Core Information Model*), definido pelo DMTF (*Distributed Management Task Force*). Enquanto o CIM tem por objetivo modelar em alto nível os elementos gerenciáveis de uma rede, o PCIM se preocupa com os aspectos específicos de políticas.

Apesar de o PCIM ser uma especialização do CIM, ele ainda é, do ponto de vista das políticas, um esquema suficientemente abstrato para não ser utilizado em plenitude, necessitando de especializações extras. Por este motivo, algumas importantes especializações foram propostas no IETF no formato de drafts. Entre elas, a principal, no contexto deste GT, é o QPIM (QoS Policy Information Model) [SNI 2003]. Apesar de importante, muitas definições do QPIM eram redundantes que outras extensões do PCIM, o que obviamente não é adequado. Como resultado, as extensões do PCIM acabavam por possuir definições redundantes entre si.

Na tentativa de amenizar esta situação, o IETF estendeu mais uma vez o PCIM através do PCIMe (*PCIM extentions*) [MOR 2003], mas agora agrupando as definições redundantes encontradas na outras extensões anteriores. Por exemplo, a definição de filtros de fluxos necessários tanto em políticas de QoS quanto de segurança, passaram a ser definidas no PCIMe.

O PCIM, PCIMe e suas extensões são definidos como um conjunto de classes e relacionamentos. Essas classes necessitam, entretanto, serem mapeadas para esquemas LDAP para

serem efetivamente utilizadas. O IETF possui um mapeamento do PCIM para LDAP [STR 2002], mas ainda não definiu um mapeamento para o PCIMe.

Outra questão importante em relação à representação de políticas no contexto do IETF é que não existe uma linguagem para expressão de políticas definida: apenas é definido o esquema de armazenamento. Isso significa também que uma linguagem para definição de políticas pode ser utilizada no nível do usuário, enquanto os esquemas LDAP do IETF seriam utilizados para armazenar tais políticas no serviço de diretórios. Mais que isso, a representação de políticas poderia ser realizada apenas através de interfaces gráficas cujos dados seriam então mapeados para o LDAP. No contexto do GT, preferimos esta última opção para representação de políticas devido aos seguintes fatores:

- Usar uma linguagem específica para criação de políticas irá obrigar os operadores a aprenderem esta nova linguagem
- Interfaces gráficas são normalmente mais intuitivas, ainda que menos poderosas em relação à capacidade de expressão das linguagens para definição de políticas
- O uso de uma linguagem também obriga a construção de parsers e validadores sintáticos e semânticos, o que acaba por aumentar a complexidade e a probabilidade de erros do software a ser desenvolvido para o piloto inicial

3.3 Suporte Preliminar às Políticas do IETF

Foi implementado no QAME um suporte preliminar às políticas do IETF através de scripts PHP que fornecem a interface gráfica do usuário, e o armazenamento de políticas em uma base LDAP que segue o mapeamento do PCIM e um mapeamento próprio do PCIMe (já que o IETF ainda não definiu um mapeamento para o PCIMe até este momento).

Deste suporte preliminar, que pode ser encontrado na URL http://noc.metropoa.tche.br pôde-se concluir:

a) Um mapeamento PCIMe para LDAP é necessário pois o mapeamento PCIM é insuficiente para representar políticas de QoS;

b) Ainda que o mapeamento PCIMe seja fornecido, outras definições complementares precisam ser criadas. Por exemplo, não é possível, usando PCIM e PCIMe determinar faixas de portas para a criação de agregados;

c) O suporte ao agendamento de políticas no PCIM e PCIMe é poderoso e permite a criação de políticas com regras temporais sofisticadas;

d) O uso de LDAP privilegia o reuso de definições de políticas. Isso melhora o uso das capacidades de armazenamento dos servidores LDAP, mas também aumenta a complexidade das interfaces de usuário;

4 Web Services

A tecnologia de Web Services [FUL 2003] pode ser conceituada, simplificadamente, como uma arquitetura para distribuição de objetos, sendo que os componentes da arquitetura são independentes de plataforma e permitem a interoperabilidade entre aplicações. O W3C possui grupos para tratar das questões de padronização de Web Services e tecnologias relacionadas, como SOAP (*Simple Object Access Protocol*), WSDL (*Web Services Description Language*) e UDDI (*Universal Description, Discovery, and Integration*).

A independência de plataforma é decorrência da adoção de XML para construção das mensagens dos protocolos usados nas comunicações. Utilizando-se XML é possível descrever dados de uma maneira estruturada, sem amarrar estas informações a tipos de dados definidos em uma determinada plataforma ou linguagem de programação [RAY 2001]. Essa independência de plataforma conduz a outra característica importante dos Web Services, que é a interoperabilidade entre aplicações. A princípio, qualquer aplicação capaz de lidar com dados XML e comunicar-se sobre um protocolo Web (como HTTP) pode ser um cliente de um Web Service, independentemente da plataforma e linguagem de desenvolvimento utilizadas na construção desta aplicação cliente e do Web Service propriamente dito.

Embora soluções com esses propósitos (independência de plataforma e interoperação), como CORBA, já existam, a tecnologia de Web Services possui algumas características próprias que a tornam interessante. Como o próprio nome dá a entender, Web Services utilizam-se de protocolos Web, os quais são padronizados e abertos (como HTTP) para realizar a troca de mensagens, ao invés de padrões binários proprietários, como RMI ou DCOM. Isso, aliado ao fato de que o entendimento desses protocolos Web está bastante difundido e que eles são suportados em, praticamente, qualquer ambiente de desenvolvimento, torna os Web Services úteis e interessantes para serem adotados como solução de distribuição. Outros pontos que, dependendo do caso, podem contar a favor dos Web Services são:

- A infra-estrutura básica necessária é composta de um servidor Web (normalmente já existe um em qualquer instituição);
- Diminuem problemas com firewalls na comunicação entre aplicações, pois se pode utilizar uma porta normalmente liberada para algum dos protocolos Web (por exemplo, a porta TCP 80 para HTTP);
- É possível implantar Web Services utilizando-se apenas software livre (por exemplo, com o servidor Web Apache com suporte à linguagem PHP);
- Web Services podem ser localizados dinamicamente, através de um esquema próprio de registro, o UDDI;
- Com relação à segurança e criptografia das mensagens, pode-se aproveitar o suporte já existente nos protocolos Web (como SSL).

4.1 A pilha de tecnologia para Web Services

Os protocolos e padrões utilizados no desenvolvimento de Web Services podem ser organizados em uma pilha, conforme ilustrado na Figure 5.

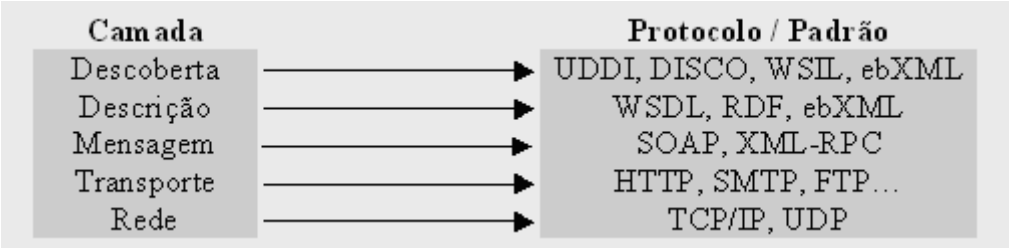

Figure 5. Hierarquia de Tecnologias Web Services

As três camadas inferiores (Rede, Transporte e Mensagem) são consideradas as camadas básicas, as quais são sempre utilizadas. É possível construir Web Services apenas com estes níveis. As camadas superiores (Descrição e Descoberta) adicionam outras funcionalidades, e são usadas dependendo das necessidades e requisitos do sistema.

4.1.1 Rede

A camada de Rede é responsável pela manipulação dos pacotes (montagem, destinação, roteamento, etc.). Como Web Services trabalham sobre a estrutura da Internet, se uma máquina já estiver funcionando em uma rede TCP/IP, o desenvolvedor não precisará preocupar-se com este nível. A infraestrutura estará pronta.

4.1.2 Transporte

Já a camada de Transporte tem uma grande importância para o desenvolvimento, pois definirá o protocolo a ser utilizado para realizar a comunicação com o Web Service. A maioria dos Web Services utiliza HTTP como protocolo de transporte. Neste caso, um servidor HTTP, como o Apache, é usado para receber e aceitar os pedidos de conexões dos clientes.

4.1.3 Mensagem

A parte mais importante da pilha está na camada de Mensagem, pois ela define o formato das instruções e documentos que serão trocados entre as aplicações. Qualquer protocolo que seja utilizado nessa camada deve ser baseado em XML, conforme está na definição de Web Services. Cabe lembrar que é, exatamente, esta característica que confere independência de plataforma e, por conseguinte, interoperabilidade aos Web Services. Os principais protocolos usados neste nível são XML-RPC e SOAP.

XML-RPC foi o primeiro protocolo para Web Services, sendo bastante simples. Ele opera sobre o protocolo HTTP, utilizando tipicamente o método HTTP POST para enviar a mensagem. O nome XML-RPC vem do fato deste protocolo possibilitar a realização de chamadas a procedimentos remotos, usando XML para enviar e receber os dados. Além disso, esse protocolo pode ser considerado como um precursor do SOAP.

Como o XML-RPC tinha problemas com questões como escalabilidade e extensibilidade, partiu-se para o desenvolvimento de um novo protocolo, no qual essas questões seriam resolvidas. Esse novo protocolo foi o SOAP, cuja versão 1.2 foi liberada como uma recomendação da W3C, em junho de 2003.

4.1.4 Descrição

A camada de Descrição possibilita descrever um Web Service. Dessa forma, baseando-se em uma descrição padronizada, pode-se aprender sobre os detalhes de um Web Service, tais como

métodos oferecidos, parâmetros de entrada e saída (com seus respectivos tipos de dados) e protocolos e padrões suportados. WSDL e RDF são os dois padrões mais populares para esta camada.

WSDL é um padrão XML para descrever um Web Service de maneira independente dos protocolos usados nas camadas de Mensagem e Transporte. Isso permite às aplicações clientes acessarem e validarem o Web Service de uma maneira bem definida. Ou seja, WSDL possibilita disponibilizar a API de um Web Service, detalhando exatamente o que ele faz, permitindo, com isso, gerar as interfaces das aplicações em tempo de execução.

RDF, por sua vez, é uma maneira para descrever objetos XML, podendo também ser usado na descrição de Web Services, através de metadados onde são descritas as operações contidas nos Web Services.

4.1.5 Descoberta

Com o uso da camada de Descoberta, pode-se organizar os Web Services num esquema de registro. Assim, organizações podem publicar a descrição de seus Web Services criados em um sistema de registro, e essas informações podem então ser pesquisadas. Os padrões usados nesta camada também são baseados em XML. Além disso, a interface de pesquisa é disponibilizada como um Web Service. Ou seja, a camada de Descoberta é constituída por um Web Service que pesquisa em um registro de Web Services e retorna à aplicação solicitante o resultado da pesquisa feita. Com isso, aplicações podem procurar, dinamicamente, pela descrição de um Web Service, fazer o download dessa descrição e criar um cliente em tempo de execução.

O padrão UDDI é um dos mais usados para esta função. Ele é implementado como um Web Service, usando SOAP para construção das mensagens, que pesquisa em repositório de metadados sobre os Web Services registrados (usualmente em um banco de dados).

4.2 Segurança dos Web Services

Hoje em dia, segurança é um item de fundamental importância em qualquer sistema. Quando esse sistema envolve distribuição e comunicação sobre uma estrutura de rede insegura (Web), essa questão torna-se ainda mais importante. O uso de Web Services permite implantar diversos aspectos relativos à segurança, tais como autenticação, políticas de acesso e criptografia, os quais podem ser usados isolados, ou em conjunto.

Identificando os usuários que acessam os serviços, pode-se estabelecer papéis, permissões ou níveis de acesso. Com isso, consegue-se fazer restrições de acesso a dados e serviços oferecidos. Para tanto, alguns esquemas já bem conhecidos, como matrizes de acesso e controle de acesso baseado em papéis, podem ser implantados.

Já o uso de criptografia impede, em tese, que pacotes, os quais porventura venham a ser capturados da rede, possam ser lidos por quem não estiver habilitado a fazê-lo. Para tanto, os pacotes transitam encriptados na rede. A maneira mais simples de utilizar criptografia é através do uso de um protocolo seguro na camada de Transporte, como, por exemplo, o HTTPS, que, por sua vez, usa SSL. Mas além do emprego de um protocolo de transporte seguro, o uso de criptografia pode ser implantado no nível de Mensagem.

4.3 Web Services e PHP

Conforme [SAR 2002], embora o PHP ainda não ofereça suporte padrão a Web Services, na forma de uma API, é possível utilizar esta linguagem para este propósito, sem maiores problemas. Existem diversas implementações de bibliotecas para SOAP e XML-RPC escritas em PHP. Ainda segundo [SAR 2002], isso levará ao desenvolvimento de uma extensão para SOAP compilada dentro

do PHP, por default. Normalmente, a principal exigência dessas bibliotecas é que o PHP esteja habilitado a manipular dados XML (através de suas API's SAX e/ou DOM). Maiores detalhes sobre o suporte a XML no PHP podem ser encontrados em [CAS 2000]. Em [SAR 2002] também podem ser encontradas referências ao uso de Web Services com outras linguagens/plataformas.

Foram feitos alguns testes e implementações simples, com três bibliotecas SOAP (de código aberto e orientadas a objetos), a fim de definir qual seria utilizada na implementação final. Essas bibliotecas são:

- PEAR::SOAP, distribuído sob a licença de uso do PHP;
- ezSOAP, distribuído sob a licença de uso GNU GPL;
- NuSOAP, distribuído sob a licença de uso GNU LGPL.

Com relação às funcionalidades, todas são semelhantes. As implementações de teste, usando estas bibliotecas, funcionaram conforme o esperado. A biblioteca NuSOAP foi a escolhida, pois a considerei como a mais simples de usar, tanto em termos de API, quanto de instalação, apesar de, segundo [FUL 03], os desenvolvedores estarem tornando PEAR::SOAP o padrão de facto para desenvolvimento de SOAP em PHP. As bibliotecas PEAR::SOAP e ezSOAP são compostas por diversos arquivos, cada uma. Enquanto isso, a biblioteca NuSOAP é composta de apenas um arquivo PHP, o qual deve ser incluído em todas as páginas onde serão utilizadas as funções e classes da API. Como o sistema de criação de proxies desenvolvido gera código PHP dinamicamente, a simplicidade de uso da API se torna fundamental.

A seguir será apresentado um código para criação de um Web Service simples e um cliente que acessa o serviço oferecido, ilustrando o uso da biblioteca NuSOAP. Esse Web Service de exemplo receberá uma string como parâmetro e retornará essa mesma string concatenada com uma outra. Por simplicidade, a biblioteca NuSOAP (nusoap.php) deve estar na mesma pasta que os dois arquivos componentes do sistema de exemplo (o cliente e o servidor). O código a seguir refere-se ao servidor. Esse arquivo deve criado com o nome soap_servidor.php.

```
1.  <?php
2.  /* Arquivo: soap servidor.php */
3.
4.  /* Inclui biblioteca NuSOAP */
5.  require_once('nusoap.php');
6.
7.  /* Instancia um servidor soap */
8.  $servidor = new soap server();
9.
10. /* Registra o serviço "teste" */
11. $servidor->register('teste');
12.
13. /* Função que implementa o serviço "teste" */
14. function teste($mensagem) {
15.     /* Testa se o parâmetro recebido é vazio */
16.     if($mensagem == '')
17.         /* Cria e retorna uma mensagem de erro */
18.         return new soap_fault('666', 'client', 'String vazia!');
19.     else
20.         /* Retorna o resultado do serviço */
21.         return "$mensagem recebido!";
22. }
23.
24. /* Inicia o processamento da requisição */
25. $servidor->service($HTTP RAW POST DATA);
26.
27. /* Assegura que mais nenhum caracter será enviado */
28. exit();
29. ?>
```

Figure 6. Cliente Web Service

O código referente ao cliente é apresentado abaixo. O nome soap_cliente.php é apenas uma sugestão.

```
1.  <?php
2.  /* Arquivo: soap cliente.php */
3.
4.  /* Inclui biblioteca NuSOAP */
5.  require once('nusoap.php');
6.
7.  /* Instancia um cliente soap */
8.  $cliente = new
    soapclient('http://noc.metropoa.tche.br/teste/soap servidor.php');
9.
10. /* Cria um array com os parâmetros a serem passados ao serviço */
11. $parametros = array('mensagem'=>'TESTE');
12.
13. /* Faz a chamada ao serviço "teste" passando o array de parâmetros */
14. $resposta = $cliente->call('teste',$parametros);
15.
16. /* Testa a ocorrência de erro na chamada ao serviço */
17. if($cliente->fault)
18.     /* Imprime mensagem de erro */
19.     print $cliente->faultstring;
20. else
21.     /* Imprime a resposta recebida do servidor */
22.     print $resposta;
23. ?>
```

Figure 7. Servidor Web Service

A saída produzida na tela, quando o cliente é acessado, pode ser conferida na Figure 8.

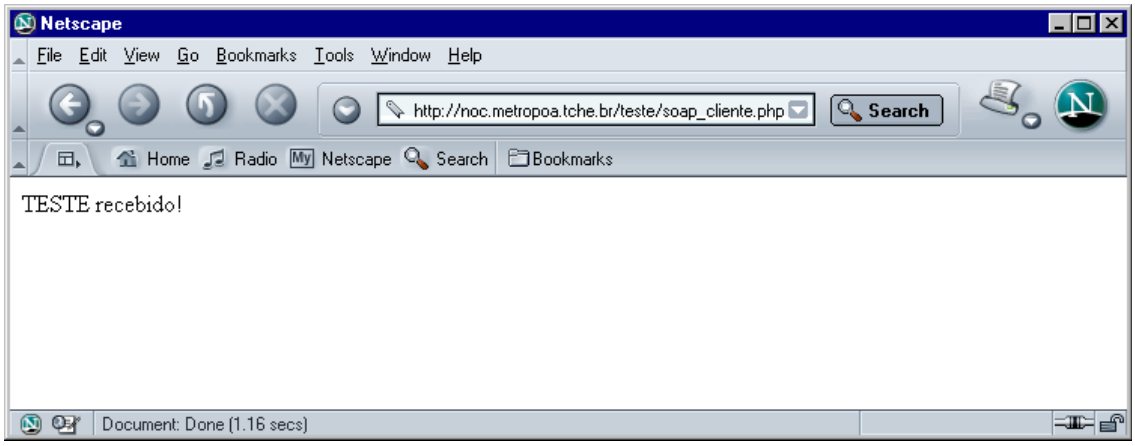

Figure 8. Resultado da consulta ao Web Service

5 Hierarquia de configurações de redes

O uso do gerenciamento de configuração baseado em políticas é interessante porque automatiza tarefas normalmente executadas manualmente pelos administradores de rede. Entretanto, a arquitetura PBNM é mais apropriada para o gerenciamento de configuração de um domínio administrativo único, controlado por uma única entidade administrativa (e.g. um administrador ou um time de administradores). Assim, apesar de as automatizações hoje suportadas no QAME serem importantes, o suporte a políticas para a configuração de dispositivos no backbone da RNP requer adaptações até hoje não encontradas em sistemas de gerenciamento de outros backbones.

A proposta do GT-Config consiste em implementar uma base de políticas distribuída através das facilidades fornecidas pelo LDAP e dos trabalhos resultantes do grupo de trabalho de diretórios (GT Diretórios) da RNP. A Figure 9 apresenta a arquitetura geral proposta.

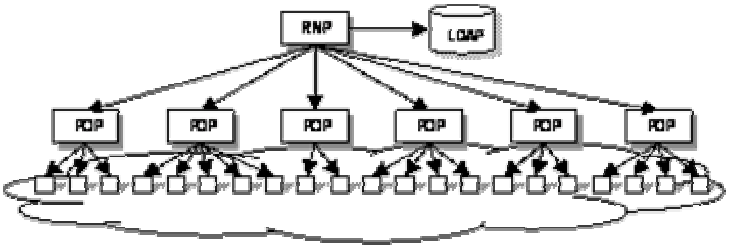

Figure 9. Proposta de arquitetura para gerência de configuração na RNP

As Políticas de QoS serão inicialmente definidas na RNP e armazenadas em uma base LDAP. Os pontos onde estas políticas serão aplicadas são também definidos pela RNP, mas informados aos diversos POPs. Cada POP é então notificado sempre que uma nova política estiver disponível no repositório. Cada administrador local de POP tem a função de aprovar a política disponível e, eventualmente, redefinir o conjunto de dispositivos finais em sua área de atuação. Após esta etapa as políticas aprovadas serão transferidas aos PDPs correspondentes aos dispositivos finais para serem implantadas na rede nos momentos agendados.

Um POP local também poderá criar suas próprias políticas e implantá-las, isoladamente, na sua rede local. Se uma política definida em um POP necessitar ser propagada a outros POPs, então a RNP seria responsável por aprovar tal política em uma instância superior, e logo a seguir notificar todos os outros POPs de interesse. O fato de se utilizar o LDAP permite uma reutilização de políticas interessante. Por exemplo, se um POP definiu uma política para videoconferência e obteve bons resultados, outros POPs podem utilizar a mesma política acessando a base de dados existente.

Na prática, o que esta sendo proposto é a inclusão de um nível extra de administração na arquitetura PBNM tradicional. Este nível extra (no caso, a RNP) tem a função de gerenciar políticas globais, aprovar políticas locais, e distribuir políticas aos POPs de interesse. Cada POP, por sua vez, seria autônomo no sentido de aceitar ou rejeitar uma política disponível no LDAP para seu contexto. Por exemplo, uma política para reserva de banda poderia ser criada pelo IMPA para uma correta transmissão de vídeo. Como a política não tem escopo apenas local, a RNP seria responsável por aprovar tal política para ser implementada no backbone da rede. Uma vez aprovada, a RNP notifica os outros POPs, e cada administrador utiliza a política em seu POP de acordo com seus interesses locais. POPs que rejeitarem a política acabarão por não configurar seus dispositivos em relação ao tráfego de vídeo gerado no IMPA, enquanto que os POPs que aceitarem tal política irão proceder com

configurações dos dispositivos de interesse, que podem incluir, por exemplo, somente um parte restrita da rede administrada por cada POP.

Assim, o sistema PBNM hierárquico permitirá que as seguintes operações relacionadas com a configuração possam ser realizadas:

(A) Operador de mais alto nível define políticas globais.

Descrição: Nesta operação um operador de mais alto nível define políticas de QoS que serão armazenadas em uma base LDAP acessível por todos os outros operadores de mais baixo nível. O operador de mais alto nível notifica a existência da nova política e os operadores de mais baixo nível decidem se tal política será ou não aplicada em seus domínios de atuação.

Exemplo: O operador de mais alto nível define uma política para privilegiar uma transmissão de aula à distância do IMPA, que será assistida em alguns POPs da RNP. O operador de mais alto nível (e.g. operador no POP-RJ) aplica tal política em seu domínio, o que gera, por exemplo, uma reserva de banda, e notifica todos os outros operadores nos POPs. Os operadores em áreas não interessadas (e.g. POP-ES) escolhem não aplicar tal política, enquanto operadores interessados (e.g. POP-RS) aplicam a política que configurará os dispositivos no seu domínio de atuação. Novamente, esse último operador notifica operadores de nível mais baixo (e.g. operadores do MetroPOA, que está ligado ao POP-RS) que mais uma vez decidem se a política será ou não aplicada.

(B) Operador de mais baixo nível propaga políticas a operadores de mais alto nível

Descrição: Nesta operação um operador de mais baixo nível que fez testes locais e convergiu em uma política ideal para uma dada situação pode desejar divulgar tal política a operadores de mais alto nível.

Exemplo: Um operador do MetroPOA cria uma política para priorização de tráfego de voz e deseja compartilhar tal política com os outros usuários da RNP (assim, outros usuários não precisaram criar e testar políticas para voz). O operador no MetroPOA armazena tal política no repositório geral e solicita a aprovação dos operadores de mais alto nível. Cada operador de mais alto nível (no exemplo, operador no POP-RS e operador no POP-RJ) pode aprovar ou rejeitar uma política por questões relacionadas com a utilização de recursos, segurança, etc.

6 Alternativas

Considerando as necessidades de automação de configuração apresentadas anteriormente, bem como as tecnologias e software existentes, as seguintes alternativas podem ser utilizadas na implementação do piloto.

6.1 Aplicação de gerenciamento

A aplicação de gerenciamento, que implementa o front-end com os administradores de rede pode ser implementada como:

a) Um software standalone que deve ser instalado nas máquinas dos administradores. O software pode ser desenvolvido em uma linguagem de programação qualquer, mas a independência de plataforma seria importante (e.g. uso de Java);

b) Um software baseado na Web que não necessita de instalação. Essa alternativa nos parece mais apropriada porque evita que o GT precise esperar pela instalação do software nos desktops dos administradores. É uma alternativa interessante porque também diminui o trabalho de suporte ao usuário no processo de instalação.

Dadas estas alternativas, preferimos utilizar a opção. Além de facilitar e agilizar o uso do sistema, implementar o piloto com software baseado na Web também significa independência de plataforma. Soma-se a isso o sistema QAME já desenvolvido, que poderá então ser adaptado para as necessidades do piloto com uma facilidade maior.

6.2 PDPs

Como apresentado na arquitetura PBNM do IETF, os PDPs são os responsáveis por traduzir as políticas de gerenciamento em ações de configuração nos dispositivos alvos. Em relação a implementação dos PDPs, as seguintes alternativas podem ser possíveis:

a) PDP como um agente SNMP. Uma forma comumente aceita de PDP é implementá-o como um agente SNMP que recebe solicitações da estação de gerenciamento, acessa as políticas de interesse a traduz tais políticas para as ações de configuração dos dispositivos alvos. O problema associado a esta alternativa é que o fluxo SNMP necessário para o acesso aos PDPs precisa cruzar os diversos domínios administrativos da RNP, o que nem sempre é possível devido a existência de firewalls. Além disso, a segurança oferecida pelo SNMPv1 (versão comumente aceita) acabaria restringindo o uso dos PDP apenas como elementos de monitoração, e não de configuração.

b) PDP como Web Service. Na tentativa de resolver os problemas intrínsecos do SNMP, o uso de Web Services poderia ser uma alternativa muito interessante na implementação de PDPs. Como os Web Services rodam sobre HTTP, o tráfego SOAP associado mais facilmente cruzaria os domínios administrativos da RNP. Além disso, o uso de protocolos seguros resolveria o problema de segurança do SNMPv1. A contrapartida no uso de Web Services é que o protocolo SOAP consome mais banda do que o SNMP. Uma alternativa a esta situação seria o uso de SOAP compactado, ainda em estudo.

Dadas as condições acima, e o domínio das tecnologias de Web Services adquirido pelos participantes do GT, opta-se então pela implementação de PDPs como Web Services a serem espalhados pelos diversos domínios administrativos da RNP que vierem a fazer parte do piloto do GT.

6.3 Representação de políticas

A representação das políticas é crítica por dois aspectos: a interação com o usuário e a tecnologia para armazenamento. A interação com o usuário deve ser, no contexto do GT-Config, muito simples para evitar que os usuários percam muito tempo definindo as políticas. A forma de armazenamento será discutida na próxima subseção.

Considerando apenas a interação com o usuário, as seguintes alternativas são possíveis:

a) Uso de uma linguagem de políticas. As linguagens para definição de políticas, como Ponder e PoP, são construídas para expressar diversos comportamentos através de sua flexibilidade. O preço pago por essa flexibilidade é, entretanto, a facilidade de uso. Os administradores devem aprender uma nova linguagem para construir as políticas de gerenciamento de redes.

b) Uso de assistentes visuais para a definição de políticas. Os assistentes visuais são recursos de interface de usuário que não forçam o administrador a aprender uma nova linguagem. Por outro lado, os assistentes são limitados e não são capazes de expressar políticas diferentes daqueles para as quais eles foram originalmente construídos.

Preferimos pela segunda opção porque, apesar da falta de flexibilidade, o uso de interfaces gráficas de usuário junto com assistentes visuais parece ter mais garantias de sucesso do que as linguagens de políticas. De nada adiantaria uma linguagem para definição de políticas flexível mas que não é utilizada por ser complexa de ser aprendida.

6.4 Armazenamento de políticas

O armazenamento de políticas pode ser feito através das seguintes opções:

a) Banco de dados. As políticas seriam armazenadas em bancos de dados e compartilhadas entre usuários do mesmo banco.

b) Serviço de diretório. As políticas seriam armazenadas em um serviço de diretórios (como o LDAP) que implementaria uma base de dados de políticas distribuída ao longo dos domínios da RNP.

c) Serviço de indexação de Web Services. As políticas seriam armazenadas em Web Services, e cadastrados no serviço UDDI.

O uso de banco de dados pode ser interessante porque os ambientes de programação suportam largamente o acesso a tais bancos através de diversas APIs. Entretanto, a implementação de uma base de dados distribuída ao longo dos domínios da RNP seria muito difícil. O uso de LDAP soluciona a questão, principalmente se considerarmos os trabalhos do GT-Diretorios. O uso de LDAP, por outro lado, é mais complexo, do ponto de vista da programação, do que o uso de bancos de dados.

O uso de UDDI não parecer ser muito apropriado ao atual momento porque se trata de uma tecnologia muito nova, e ainda não totalmente aceita. Como o próprio IETF recomenda o uso de LDAP a ponto de já ter definido mapeamentos do PCIM para LDAP, consideramos que o pilo do GT-Config também deve ser baseado em LDAP.

7 Conclusões

Consideramos que o gerenciamento de configuração é uma necessidade que irá se tornar cada vez mais freqüente em backbones de alta velocidade com suporte a QoS como a RNP e a rede Giga. Neste relatório foi apresentada a arquitetura de gerenciamento baseado em políticas do IETF, bem como o ambiente de gerência de QoS QAME que foi construído respeitando tal arquitetura. A proposta do trabalho consiste na adaptação do QAME de forma a introduzir um nível administrativo extra à arquitetura PBNM do IETF, como apresentado.

No nosso ponto de vista, os esforços administrativos futuros em relação as configuração para suporte a QoS não poderão mais ser suportados através das ferramentas atualmente utilizadas pelos administradores de POPs, que muitas vezes se restringem a intervenções locais nos dispositivos da rede. Tais sistemas não serão capazes de suportar uma intensa e rápida reconfiguração agendada de dispositivos, o que os tornará insuficientes para o gerenciamento de QoS no contexto da RNP (esta constatação, obviamente, pode ser estendida a outros grandes *backbones* da mesma forma). Assim, acreditamos que um esforço no gerenciamento de configuração é necessário desde já.

A proposta, como colocado, sugere a utilização de QAME, por este já ser desenvolvido especificamente para o gerenciamento de QoS, mas o ponto mais importante, entretanto, nos parece ser a disponibilização de um serviço de diretórios LDAP a toda a RNP (tarefa relacionada com o GT-Diretórios) e a automatização de ações de configuração, tema ainda não abordado nos GTs de 2002/2003. Complementarmente, a definição de políticas de QoS por parte da RNP e dos administradores do POPs é função direta do conhecimento do comportamento da rede e dos requisitos de QoS de cada aplicação de interesse.

Em relação ao conjunto de tecnologias a serem utilizadas, dadas as alternativas apresentadas, consideramos que o piloto deverá se basear fortemente em:

- Web Services para a implementação dos PDPs;
- Interface gráfica baseada na Web do QAME;
- Definição de políticas através de assistentes gráficos;
- Armazenamento de políticas em bases LDAP seguindo os mapeamentos do IETF para o PCIM.

8 Referências

[CAS 2000] CASTAGNETTO, J.; SCHUMANN, S.; RAWAT, H.; SCOLLO, C.; VELIATH, D. T. Professional PHP Programando. 1. ed. São Paulo: Makron, 2000.

[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.

[CLA 2000] CLARK, R. The Mechanics of Policy-Based Management. Network Magazine, p.44-41, Mar. 2000.

[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.

[EXT 2001] EXTREME NETWORKS. EPICenter 3.0 Technical Specification. Disponível em: <http://www.extremenetworks.com/products/prod pdf/EPICenter.pdf>.

[FUL 2003] FULLER, J.; FUECKS, H.; EGERVARI, K.; WATERS, B.; SOLIN, D.; STEPHENS, J.; REYNOLDS, L. Professional PHP Web Services. 1st ed. Birmingham, UK: Wrox, 2003.

[GRA 2001] GRANVILLE, L.Z.; CECCON, M.B.; TAROUCO, L.M.R.; ALMEIDA, M.J.B. NetPlus – Um Ambiente para Gerência de QoS baseado na Web. NewsGeneration, vol. 5, nr. 4. RNP, Julho de 2001.

[HAL 2003] HALPERN, J.; ELLESSON, E. Policy Framework (policy) IETF Working Group. Disponível em: <http://www.ietf.org/html.charters/ policy-charter.html>.

[HEW 2003] HEWLETT-PACKARD. HP OpenView Homepage. Disponível em: <http://openview.hp.com>.

[HEW 2001a] HEWLETT-PACKARD. HP OpenView PolicyXpert Homepage. Disponível em: <http://www.openview.hp.com/products/policyexpert/>.

[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.

[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>.

[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.

[MOR 2001] MOORE, B.; ELLESSON, E.; STRASSNER, J.; WESTERINEN, A. Policy Core Information Model - Version 1 Specification. IETF Request For Comments 3060, February 2001.

[MOR 2003] MORRE, B. Policy Core Information Model (PCIM) Extensions. IETF Request For Comments 3460. January 2003.

[RAY 2001] RAY, E. T.; MADEN, C. R. Learning XML. 1st ed. Sebastopol, USA: O'Reilly & Associates, 2001.

[SAR 2002] SARANG, P.; BROWNE, C.; AYALA, D.; CHOPRA, V. Professional Open Source Web Services. 1st ed. Birmingham, UK: Wrox, 2002.

[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 2003] SNIR, Y.; TAMBERG, Y.; STRASSNER, J.; COHEN, R.; MOORE, B. Policy QoS Information Model. IETF draft <draft-ietf-policy-qos-info-model-05.txt>, May 2003.

[STR 2002] 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-16.txt> work in progress).

[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.