Network Executive: Implementação de uma Arquitetura para Substituição Automática de Políticas em Sistemas PBNM

Gustavo Augusto Faraco de Sá Coelho, Maria Janilce Bosquiroli Almeida, Liane Margarida Rockenbach Tarouco, Lisandro Zambenedetti Granville

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{gcoelho, janilce, liane, granville}@inf.ufrgs.br

Abstract. *This paper describes an architecture able to support automatic policy replacement when defined network events occur. Thus, an automated system capable to react to network behavior changes is achieved, deploying a new policy better suited to this new network behavior. The architecture main objective is to help network managers with network operations maintenance task when the occurrence of special events leads to network malfunction. A formal architecture able to provide this functionality and new related concepts are presented, as well as a functional prototype of this architecture.*

Resumo. *Este artigo descreve uma arquitetura capaz de suportar a substituição automática de políticas quando determinados eventos ocorrem na rede. Desta forma, obtém-se um sistema autônomo capaz de reagir a mudanças no comportamento da rede, aplicando uma nova política mais adequada a esta nova situação em que a rede se encontra. O objetivo principal desta arquitetura é auxiliar o gerente na tarefa de manutenção da operação de sua rede quando da ocorrência de eventos que possam perturbar o seu funcionamento. A arquitetura genérica capaz de prover esta funcionalidade e os novos conceitos envolvidos são apresentados, bem como a implementação desta arquitetura em um sistema de gerência.*

1. Introdução

Observando o mercado atual de sistemas de gerenciamento de redes, nota-se a consolidação dos sistemas de Gerenciamento de Redes Baseado em Políticas *(PBNM – Policy Based Network Management)* [Sloman 94] como uma alternativa interessante aos sistemas de gerenciamento tradicionais [Shepard 2000]. Quando surgiu, o paradigma PBNM prometia ser uma solução eficaz contra o problema enfrentado pela maioria dos gerentes de redes: a crescente heterogeneidade e complexidade dos novos dispositivos que eram adicionados às suas redes.

Contudo, assim como outras tecnologias recém desenvolvidas, os primeiros sistemas PBNM não foram capazes de atingir todos os objetivos propostos e exigidos pelos usuários [Conover 99]. Problemas como a falta de compatibilidade entre os sistemas de gerência e os dispositivos de rede e as deficiências de interoperabilidade entre soluções de fornecedores distintos reduziram a velocidade de adoção desses sistemas pelas organizações [Boardman et al. 2002A].

Atualmente, entretanto, observa-se uma nova geração de sistemas PBNM que, se não tiveram todas as suas deficiências solucionadas, pelo menos já apresentam melhores

resultados e, como conseqüência, impulsionaram a adoção dessas ferramentas no cotidiano dos gerentes de redes [Boardman et al. 2002]. Avaliando as opções disponíveis no mercado, é natural que haja diferentes funcionalidades em cada ferramenta [Coelho 2001]. Entretanto, uma análise mais atenta permite que seja observado que não há mecanismos para automatização da aplicação das políticas. Se a possibilidade de agendar uma política para ser aplicada em determinado momento é uma opção existente em algumas ferramentas hoje, também é verdade que essa operação é problemática, já que as condições da rede neste momento futuro podem ser incertas. O que se observa, atualmente, é o uso de uma ferramenta auxiliar de monitoração para avaliar a eficácia da aplicação da política feita pelo sistema PBNM. A monitoração, a avaliação dos dados coletados e a decisão pela manutenção ou substituição da política são tarefas realizadas manualmente pelo gerente da rede. Além de sobrecarregar o gerente com mais tarefas, este processo pode sofrer atrasos e ser conduzido de forma errônea por depender da intervenção humana.

Este artigo apresenta uma solução para essa necessidade de automatização na aplicação de políticas. Além disso, a solução leva em consideração essa incerteza quanto ao correto momento em que a política deve ser colocada em operação. Nota-se que essa incerteza é referente ao momento no tempo em que a política deve ser aplicada e não à situação onde ela deve ser aplicada. Isto ocorre porque o gerente da rede deve ter a noção de quais políticas devem estar em operação quando determinadas situações / problemas são enfrentados pela sua rede. Entretanto, é difícil, se não improvável, que o gerente saiba em que momento no futuro tais situações / problemas vão ocorrer para que ele possa agendar a aplicação da política utilizando as ferramentas atuais.

O trabalho a seguir está dividido da seguinte forma: a próxima seção apresenta uma breve revisão dos conceitos envolvidos com o PBNM. A terceira seção introduz a solução proposta neste trabalho, descrevendo conceitos envolvidos e a arquitetura que dá suporte ao processo de substituição. A quarta seção apresenta a validação desta proposta através da implementação de um sistema PBNM com suporte a esta tarefa. Por fim, algumas conclusões atingidas e propostas para futuros trabalhos são apresentadas na última seção.

2. Gerenciamento de Redes Baseado em Políticas (PBNM)

O gerenciamento de redes baseado em políticas utiliza o conceito de políticas para que o gerente da rede possa definir o que ele quer, utilizando para isso uma linguagem mais abstrata do que a utilizada pelo gerenciamento tradicional. Outro grande benefício é a reutilização de uma mesma política definida pelo usuário em vários dispositivos da rede. Com isso, a tarefa de configuração da rede é facilitada, assim como a existência de um pequeno conjunto de políticas torna a manutenção delas mais rápida. Por fim, a utilização de uma linguagem abstrata para definir as políticas resulta em certo grau de independência em relação aos fornecedores de dispositivos, assim como das tecnologias implementadas por eles.

Uma das características que tornam o PBNM diferente do gerenciamento tradicional é a forma como o gerente interage com o sistema. Enquanto que as ferramentas tradicionais mostram detalhes particulares de cada dispositivo, o PBNM utiliza o conceito de políticas *(Policy Rules)* para abstrair estas particularidades, sem que isso resulte na perda da capacidade de gerenciamento. Basicamente, uma política é um par de condição e ação, onde uma ou mais ações são executadas quando uma ou mais condições são satisfeitas. Normalmente, ela é definida através de uma clausula *if* -

then, na forma de *if* <*condição*> *then* <*ação*>. A expressão <*condição*> pode ser simples ou composta e pode estar relacionada com elementos como servidores, aplicações, protocolos, usuários, etc. Já a expressão <*ação*> especifica quais serviços são permitidos ou negados, quais parâmetros são alterados, etc, quando <*condição*> for verdadeira. O conjunto de ações que devem ser tomadas pode estar ordenado ou não ordenado. É possível que uma política consista de outras políticas, tomando uma forma hierárquica que é essencial para que políticas complexas sejam construídas a partir de política mais simples. Por ter esta capacidade de expressão, o uso de políticas torna o gerenciamento mais simples e fácil.

O IETF definiu também um modelo para sistemas PBNM [Mahon et al. 2000]. De forma simplificada, um sistema PBNM deve ser capaz de fornecer, no mínimo, funcionalidades para permitir que o usuário possa definir e atualizar as políticas, ter funções para armazenar e recuperar as políticas e ser capaz de interpretar, implementar e aplicar as políticas. Os elementos da arquitetura PBNM que possibilitam estas funcionalidades são ilustrados na figura 1.

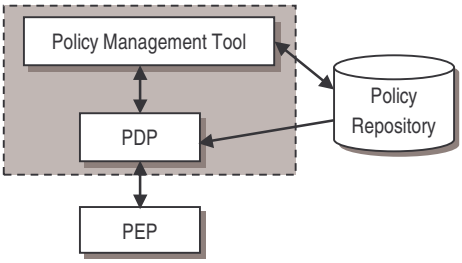

Figura 1. Arquitetura PBNM

Nesta arquitetura, o *Policy Management Tool* é o elemento que permite uma entidade (o gerente da rede ou uma outra aplicação, por exemplo), definir e atualizar as políticas, e opcionalmente, monitorar a aplicação delas. É a interface gráfica e/ou os *scripts* que permitem a interação entre o usuário e o sistema. O PDP (*Policy Decision Point*) é o elemento responsável por recuperar e distribuir as políticas para serem aplicadas nos PEPs (*Policy Enforcement Points*). Ele pode, opcionalmente, traduzir as regras para uma linguagem que possa ser interpretada pelo PEP. O PEP é a entidade cujo comportamento é definido pela política. Ele é o objetivo-alvo da política. Por último, há o Repositório de Políticas *(Policy Repository),* o qual possibilita o armazenamento permanente e a recuperação das políticas.

3. Policy of Policies

Quando uma política deve ser implantada na rede, o PDP inicia a tradução dela em ações que possam ser executadas pelo tipo específico de PEP que está sendo configurado. Durante este processo de aplicação da política, podem ocorrer falhas devido a conflitos entre a nova política e políticas previamente implantadas, falta de recursos ou indisponibilidade do dispositivo, entre outras razões. Em todos estes casos, o PDP deve notificar o gerente da rede sobre o fracasso do processo de aplicação da nova política. Em contrapartida, se a política é considerada implantada com sucesso, nenhuma verificação adicional será feita pelo sistema PBNM, repassando para o gerente a responsabilidade de verificar a eficiência da política após sua implantação. Nestes casos, é comum o gerente utilizar ferramentas adicionais de monitoração para obter dados suficientes para ele concluir se a política está atingindo seus objetivos ou não.

Quando um problema na política recém implantada é detectado, é provável que as políticas inapropriadas sejam substituídas ou removidas, sendo novas políticas selecionadas e aplicadas. Todo este procedimento de substituição de políticas visa trazer a rede para um estado consistente e correto de operação. Como já mencionando, atualmente este processo é todo executado manualmente pelo gerente da rede. Algumas ferramentas disponibilizam uma opção de agendar a aplicação de uma política em um determinado instante de tempo. Entretanto, como é improvável que o gerente saiba com antecedência quando um problema irá ocorrer, soluções deste tipo tornam-se pouco úteis e eficientes.

A solução proposta a seguir automatiza estes procedimentos com o objetivo de auxiliar o gerente na manutenção da operação correta da sua infraestrutura de rede. Esta solução tenta resolver os problemas decorrentes da detecção de falha em uma política após sua implantação na rede. Para isto, um novo conceito, denominado de "Política de Políticas" *(Policy of Policies – PoP)* foi desenvolvido. Este novo conceito, juntamente com a arquitetura que dá suporte a ele, são as bases para a implementação do sistema que é abordado ao longo deste trabalho e que visa automatizar o processo de substituição de políticas em sistemas PBNM. Este conceito será abordado a seguir.

3.1. Conceitos

O conceito de *Policy of Policies (PoP)* tem como objetivo principal permitir a criação de um novo tipo de política capaz de coordenar a implantação e a aplicação de outras políticas. Este conceito é diferente, mas complementar à noção de políticas hierárquicas previsto pelo IETF, onde políticas complexas são criadas a partir de políticas mais simples. É importante ressaltar que uma PoP não é uma política complexa composta de políticas mais simples e sim um mecanismo que permite a automatização do processo de substituição de políticas. A fim de evitar esta confusão, será visto inicialmente o trabalho desenvolvido pelo IETF que prevê políticas complexas sendo construídas pela junção de políticas mais simples.

Composição de políticas

Como mencionado por Westerinen et al. [Westerinen et al. 2001], uma política é uma sentença na forma "*if* <*condição*> *then* <*ação*>". Ela representa a relação entre um conjunto de ações (... *then* <*ação*>) que devem ser executadas sempre que um conjunto de condições (*if* <*condição*> ...) for considerado verdadeiro. Assim, uma política deste tipo forma a base para a construção de políticas mais complexas sempre que forem agrupadas duas ou mais políticas como a descrita. É este processo de composição de políticas que resulta na definição de uma política considerada mais complexa. A decisão de agrupar várias políticas com objetivos comuns em uma única política mais complexa objetiva facilitar a tarefa de manutenção do conjunto de políticas.

O processo de implantação desta política exige que o sistema PBNM avalie o trafego da rede contra a primeira regra. A seguir, o mesmo fluxo de dados é comparado com a segunda regra, e assim por diante. Quando o trafego que está sendo analisado satisfizer a regra que está sendo avaliada, a ferramenta PBNM deve executar as ações definidas naquela regra e o processo de avaliação da política é encerrado. A aplicação da técnica mencionada no processo de avaliação de uma política similar à apresentada, evita a execução de ações distintas de regras diferentes sobre um mesmo fluxo de dados, o que resultaria em uma situação de conflito. Neste caso foi utilizado o conceito de

prioridades para resolver o problema de resolução de conflitos, com políticas de mais alta prioridade sendo escolhidas em detrimento de outras com prioridade inferior. Outras técnicas para detecção e resolução de conflitos são discutidas por Lupu et al. [Lupu et al. 99].

O conceito mais importante deste exemplo é o que permite que uma política seja constituída de outras políticas, permitindo a definição de políticas complexas a partir de outras mais simples [Moore et al. 2001] [Moore 2002].

Policy of Policies

A substituição de políticas quando as características de QoS observadas são muito diferentes das características especificadas, é uma tarefa executada manualmente pelo gerente da rede. O conceito de *Policy of Policies* foi desenvolvido, pois a automação de tal processo de substituição poderia facilitar a tarefa de manutenção da operação da rede.

Como já mencionado, a substituição de uma política depende do plano de ações da rede. Diferentes redes possuem diferentes estratégias de substituição de políticas. Desta forma, um dos requisitos para a automação da substituição de políticas é a definição de quais políticas devem substituir outras, e sob quais circunstâncias. Estas definições são baseadas nos objetivos de operação e desempenho da rede, e podem ser vistas como políticas, mas em um nível funcional superior as das políticas tradicionais.

Neste contexto, o conceito de *Policy of Policies* ou PoP é introduzido. O conceito de PoP é similar ao de meta-políticas [Koch et al. 96], já que ele define políticas de mais alto nível. Entretanto, uma PoP pode ser considerada uma meta-política que coordena a aplicação e a substituição de políticas tradicionais de QoS quando eventos especiais ocorrem. Estes eventos especiais são tipicamente gerados quando problemas nas políticas previamente aplicadas são observados. É importante mencionar que apesar deste trabalho utilizar PoPs para coordenar políticas de QoS, o conceito de PoP pode ser empregado com outros tipos de políticas, como as utilizadas para controle de acesso e segurança.

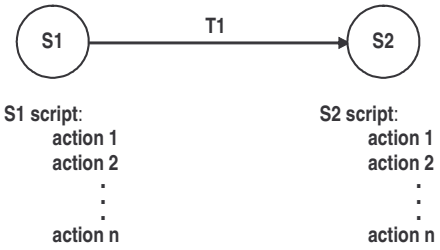

Figura 2. Definição de uma PoP genérica.

A definição de uma PoP requer referências para todas as possíveis políticas que podem ser aplicadas em um dispositivo. Este conjunto de referências inclui tanto as políticas que são usadas diretamente quanto às definidas para serem utilizadas em substituição a outras políticas. Além disso, a definição de uma PoP também requer a identificação dos eventos que podem acionar a substituição das políticas. Para isto, uma máquina de estado finito é utilizada como forma de definir e representar uma PoP. Neste formalismo, os nodos representam *scripts* que são executados quando da substituição das políticas enquanto que as transições representam os eventos que levam à substituição da política.

A figura 2 apresenta a definição de uma PoP genérica representada através de uma máquina de estado finito. A máquina é composta de um conjunto de nodos e um conjunto de arcos. Cada arco é composto por um par de nodos e é usado para ligar estes dois nodos. Cada arco representa uma transição de um nodo para outro, de forma que o conjunto de arcos também pode ser definido como um conjunto de transições. A partir da PoP definida na figura anterior, tem-se que:

```
FSM = { N(FSM), T(FSM) }, onde
      N(FSM) = { S1, S2 }
      T(FSM) = { (S1,S2) }
```

Além de um conjunto de nodos e de um conjunto de transições, uma PoP também requer um conjunto de *scripts*, onde cada *script* está associado a um nodo. Em uma PoP, cada nodo representa um estado esperado que a rede pode assumir. Uma transição da máquina de estado finito é acionada quando um evento especial ocorrer. Neste caso, o nodo atualmente ativo é deixado e um novo nodo é alcançado. Assim, uma transição leva indiretamente à execução de um *script* de substituição de políticas associado ao nodo recém alcançado. Já que uma transição representa a ocorrência de um evento, a máquina de estado finito irá trocar seu estado somente quando um evento acontecer na rede. Os possíveis estados que a rede pode assumir e os possíveis eventos que a mesma rede pode gerar são informações especificas àquela rede. Assim, o gerente da rede é a pessoa mais recomendada para definir PoPs que automatizem a substituição de políticas associadas a sua rede.

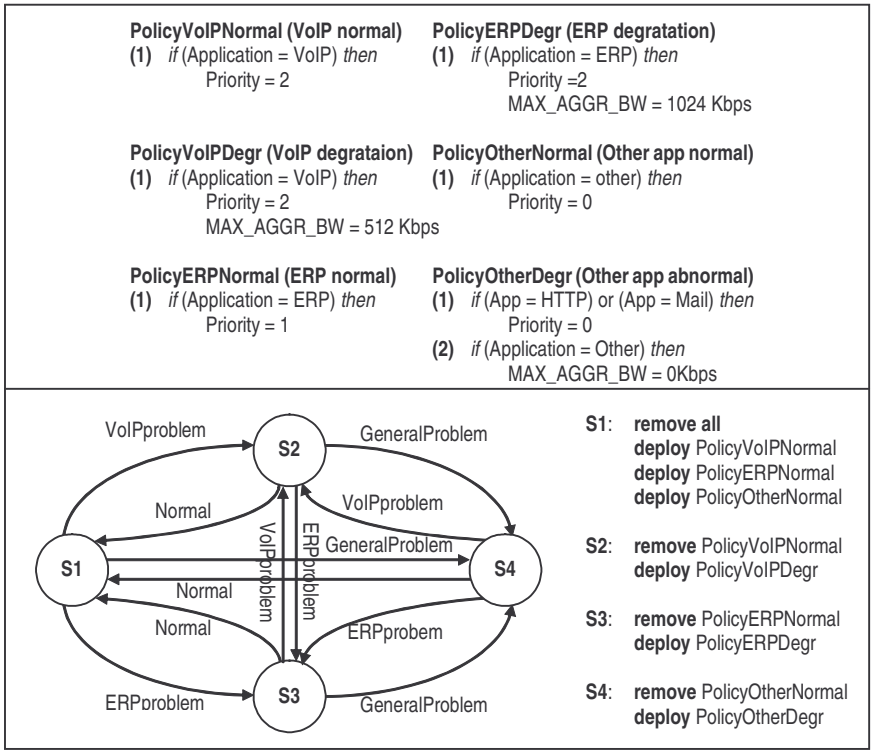

Figura 3. Políticas e PoP correspondente

Quando do planejamento de uma política, o gerente da rede pode agrupar as políticas de acordo com o plano de ações da sua rede. Já quando da presença de PoPs, é esperado que o gerente defina as políticas e as PoPs (com suas máquinas de estado e *scripts* de

substituição). É importante, neste caso, planejar as políticas e as PoPs juntas, a fim de alcançar uma estratégia de substituição de políticas adequada. Políticas não devem ser definidas isoladamente, mas sim como um grupo e de acordo com a estratégia geral de gerenciamento da rede. Neste cenário, a utilização de PoPs pode ser uma boa ferramenta para auxiliar aos gerentes na tarefa de traduzir os objetivos do seu plano de gerenciamento, expresso em linguagem de alto nível, para uma outra linguagem que pode ser usada para gerenciar suas redes.

Para finalizar esta seção será apresentada a definição completa de uma PoP, incluindo a máquina de estado finito, os scripts de substituição e as políticas envolvidas. Baseado em um cenário hipotético, o conjunto mais adequado de políticas e a correspondente PoP seriam as apresentadas na figura 3.

3.2. Arquitetura

Nesta seção será apresentada a proposta de uma arquitetura capaz de suportar os conceitos associados à PoP apresentados nas seções anteriores. Nesta proposta serão descritos os elementos presentes na arquitetura, suas características e funções, bem como as relações funcionais existentes entre eles. Não haverá preocupação com detalhes de implementação visto que estes serão abordados nas próximas seções deste trabalho.

O ponto principal da arquitetura proposta prevê a utilização de um mecanismo de substituição automática de políticas dentro dos consumidores de políticas (PDPs). O modelo desta proposta é apresentado na figura a seguir e estende as definições do IETF sobre sistemas de Gerenciamento de Redes Baseado em Políticas vistas anteriormente.

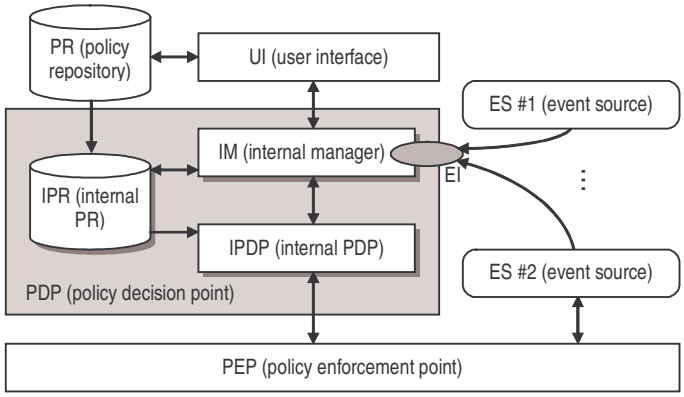

Figura 4. Arquitetura PoP

O gerente da rede utiliza a UI *(User Interface)* para definir tanto políticas quanto PoPs. Estas políticas e PoPs são então armazenadas em um repositório de políticas externo *(PR – Policy Repository)*. Este repositório tem as mesmas características e funções do repositório definido pelo IETF, exceto pelo fato que na arquitetura proposta ele também armazena PoPs. O gerente pode recuperar informações previamente definidas para alterá-las ou removê-las. Para definir o comportamento da rede, o gerente deve contactar os PDPs através da interface gráfica (UI). Estes PDPs então interagem com os PEPs para implantar as políticas. Elementos denominados de *Event Sources* (ES) podem notificar PDPs quando situações especiais são detectadas na rede. Um exemplo típico de um destes elementos seria um monitor de QoS que verifica parâmetros de QoS da rede e gera notificações sempre que degradações de QoS forem detectadas [Ribeiro et al. 2001]. *Event Sources* notificam os PDPs através da interface disponibilizada por outro elemento denominado de *Event Interface* (EI). Este elemento é caracterizado como o único ponto de acesso para a comunicação de eventos existente nos PDPs.

A arquitetura proposta divide o PDP originalmente definido pelo IETF em três elementos básicos: o IPR *(Internal Policy Repository)*, o IM *(Internal Manager)* e o IPDP *(Internal PDP)*. O IPDP age exatamente da mesma forma que o PDP padrão definido pelo IETF age. Ele é controlado por um elemento gerenciador (neste caso o IM), recupera políticas de uma base de dados (neste caso o IPR) e contata PEPs para proceder com as tarefas de implantação e remoção de uma política. As políticas são utilizadas pelo IPDP nos processos de aplicação e remoção de política, enquanto que as PoPs são utilizadas pelo IM para controlar a estratégia de substituição de políticas dentro do IPDP. O IPR é funcionalmente idêntico ao repositório de políticas externo (PR) e, assim, armazena o mesmo tipo de informação: políticas e PoPs. Entretanto, a interface de comunicação do IPR não deve ser baseada no protocolo LDAP, já que é provável que esta base de dados será implementada em memória volátil.

4. Network Executive

A arquitetura proposta na seção anterior pode ser considerada um modelo genérico para prover um mecanismo de substituição automática de políticas. Isto pode ser afirmado, pois na apresentação da arquitetura mencionada não foram considerados aspectos relacionados com a implementação da mesma, apenas foram especificados os elementos que compõem a arquitetura, suas funcionalidades e as relações existentes entre eles.

Esta seção objetiva descrever o trabalho envolvido com a implementação concreta deste modelo apresentado anteriormente. Assim, um sistema de gerenciamento baseado em políticas, desenvolvido em trabalhos anteriores e denominado Network Executive [Coelho et al. 2001A], teve sua arquitetura expandida para suportar o mecanismo proposto neste artigo. As próximas seções descrevem a nova arquitetura deste sistema, as MIBs definidas para suportar a configuração de políticas e PoPs dentro dos PDPs e o esquema de dados desenvolvido para o armazenamento das informações em um diretório LDAP.

4.1. Arquitetura

A arquitetura atual do sistema Network Executive é resultante da combinação da sua arquitetura previamente implementada com o modelo genérico proposto na terceira seção deste artigo. A figura 5 ilustra esta arquitetura, apresentando os elementos, suas relações e os protocolos envolvidos na comunicação.

Observando-se a figura mencionada é possível perceber que muitos dos elementos apresentados possuem um correspondente direto no modelo da figura 4 e na arquitetura proposta pelo IETF (figura 1). Assim, o elemento denominado PEP é a mesma entidade apresentada nas arquiteturas citadas anteriormente, e portanto, possui as mesmas características e funcionalidades. A configuração do PEP é feita pelo PDP, como já mencionado em seções anteriores, através de um ou mais protocolos suportados tanto pelo PDP quanto pelo PEP. Na figura 5 esta comunicação é feita por Telnet, SNMP ou COPS, por exemplo.

O elemento PDP exibido na figura anterior engloba as entidades IPDP, IM e IPR apresentadas na arquitetura genérica com suporte a PoPs. Assim, é este elemento que tem a responsabilidade de carregar as políticas e as PoPs de um repositório externo, na figura identificado por “Repositório LDAP”, construir em memória local a representação da máquina de estados finita definida pela PoP e aplicar as políticas de

acordo com o estado vigente. Este elemento utiliza o protocolo LDAP para acessar o diretório com as informações relevantes e que estão armazenadas de acordo com o esquema de dados que será discutido nas próximas seções.

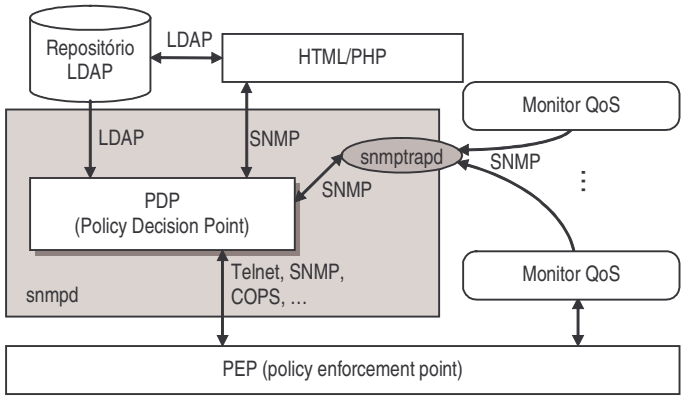

Figura 5. Arquitetura Network Executive

O PDP foi implementado como um módulo dinâmico do serviço de SNMP do Linux, conhecido como NET-SNMP, e representado na figura por *snmpd*. A execução deste módulo ocorre de forma similar a de uma biblioteca dinâmica (DLL) do ambiente Windows. Neste caso, o módulo é carregado pelo serviço *snmpd* na sua inicialização e torna-se responsável pelo tratamento de todos os objetos de uma MIB cadastrada pelo módulo junto ao serviço *snmpd*. Este cadastro é feito informando ao serviço o objeto inicial da MIB desejada. Todos os objetos contidos a partir deste nodo inicial são de responsabilidade deste módulo. A MIB e os objetos definidos serão discutidos na próxima seção.

Por estar contido em um serviço SNMP, o PDP implementado pelo sistema Network Executive recebe informações da interface com o usuário (elemento UI da figura 4) através de comandos SNMP. Esta interface gráfica está codificada através de páginas HTML e scripts PHP, exigindo apenas um servidor HTTP e clientes com um navegador Web. O servidor HTTP não precisa ser executado na mesma estação que contém o serviço de diretório LDAP nem na estação que contém o serviço *snmpd* e o PDP. Observa-se que desta forma obtém-se um bom grau de escalabilidade do sistema, requisito importante em sistemas de gerenciamento de redes.

Finalizando a apresentação da arquitetura Network Executive, é importante mencionar a utilização do serviço *snmptrapd*, também contido no pacote NET-SNMP, e que é responsável por receber os eventos externos gerados pela rede. O *snmptrapd* possui assim o papel do elemento EI apresentado na figura 4, receber eventos externos e repassá-los para o elemento IM contido no PDP, o qual é responsável por controlar o mecanismo de avaliação da máquina de estado finita e proceder com a substituição de políticas caso necessário. Este serviço recebe *traps* SNMP e executa um programa externo responsável por traduzir esta *trap* em comandos SNMP-SET da MIB contida no PDP. A escrita nestes objetos SNMP resulta na execução da rotina que avalia o evento recebido e decide pela substituição ou não de uma política. Observa-se que a implementação do elemento EI através de um serviço de recepção de *traps* permite que qualquer elemento de monitoração da rede possa servir de fonte de geração de eventos para o mecanismo de substituição de políticas, desde que este seja capaz de enviar *traps* SNMP. Na implementação atual do sistema Network Executive são utilizados os monitores de QoS definidos por Ribeiro et al.[Ribeiro et al. 2001], por exemplo.

4.2. MIB - PoP

Como já mencionado na seção anterior, o elemento PDP da arquitetura Network Executive é configurado através de comandos SNMP. Portanto, ele implementa uma MIB, desenvolvida especialmente para este sistema. Esta MIB é dividida em vários ramos, havendo grupos para verificação do estado de operação do PDP *(consumerStatus)*, para configuração da sua operação *(consumerConfig)*, para recuperação de estatísticas sobre a aplicação das políticas e PoPs *(consumerStatistics)*, entre outros. Esta seção abordará apenas os grupos mais importantes desta MIB, os quais influem diretamente no funcionamento do mecanismo de substituição das políticas. São estes os grupos que estão relacionados na figura 6.

Esta figura apresenta três grupos pertencentes à MIB – PoP, o grupo *consumerEvents*, o *consumerPolicies* e o *consumerPoPs*. O primeiro é responsável pela recepção dos eventos gerados pela rede e pela ativação do mecanismo que avalia a necessidade ou não da substituição das políticas. O segundo e o terceiro grupos são utilizados para comandar a recuperação de políticas e PoPs do repositório para o PDP. Pode-se observar, também, nesta figura os outros grupos pertencentes a esta MIB e já mencionados anteriormente, mas que não serão abordados em profundidade neste trabalho.

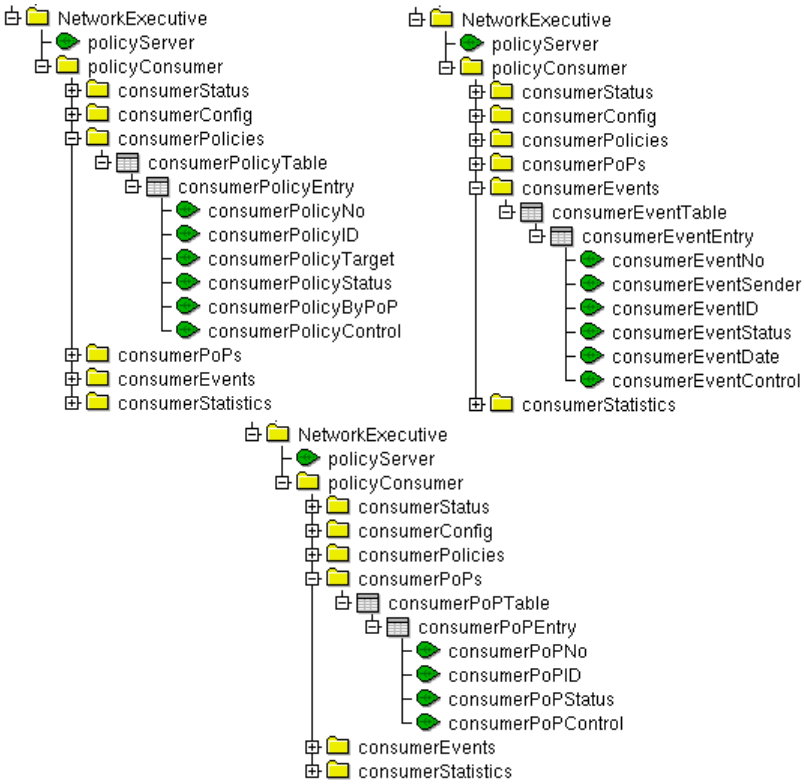

Figura 6. Management Information Base (MIB) - PoP.

O grupo *consumerPoPs* é composto de uma tabela utilizada tanto para comandar a recuperação de uma nova PoP quanto para visualizar quais PoPs estão ativas em determinado PDP. O atributo *consumerPoPNo* é um inteiro positivo e consecutivo que serve como chave primária para esta tabela, é um atributo apenas de leitura e é controlado pelo próprio PDP. *consumerPoPID* é a referência que o PDP tem para recuperar a PoP no repositório. Como esta arquitetura utiliza um diretório LDAP, esta

referência é codificada como um '*distinguished name*'. Através desta referência, o PDP pode acessar o diretório e recuperar a PoP correta. O atributo de leitura *consumerPoPStatus* serve para o usuário identificar o estado de uma determinada PoP (*active*, *under construction*, *failure*, entre outros). Por último o atributo *consumerPoPControl* é utilizado para inicializar uma nova PoP, criando uma nova linha na tabela '*consumerPoPTable*'. Seu funcionamento é similar às variáveis de controle definidas na MIB RMON. Quando se deseja criar uma nova PoP, este atributo é configurado com um determinado valor, permitindo que o atributo *consumerPoPID* seja definido e/ou alterado. Para ativar esta nova PoP, o atributo é configurado para outro valor definido, informando ao PDP que esta PoP deve ser recuperada do repositório e ativada. O processo de remoção de PoPs armazenadas no PDP também é comandado através deste objeto.

O grupo *consumerPolicies* possui atributos com funções similares aos objetos definidos no grupo descrito anteriormente. Assim, o atributo *consumerPolicyNo* é funcionalmente igual ao objeto *consumerPoPNo*. Esta relação é verdadeira também para os atributos *consumerPolicyID*, *consumerPolicyStatus* e *consumerPolicyControl* que possuem as mesmas características que os objetos *consumerPoPID*, *consumerPoPStatus* e *consumerPoPControl*, respectivamente. O atributo *consumerPolicyTarget* possui uma referência para o PEP associado a esta política, codificado como um '*distinguished name*'. O valor deste atributo é carregado automaticamente pelo PDP quando uma política é ativada. Este valor é recuperado através das informações contidas no repositório, o qual contém a relação de quais políticas devem ser aplicadas em quais PEPs. Observa-se assim que é possível notar a ocorrência de uma mesma política em vários PEPs distintos através da consulta desta tabela da MIB - PoP. Já o objeto *consumerPolicyByPoP* é utilizado para observarmos quais políticas estão associadas a quais PoPs. Este atributo contém um valor inteiro que serve como referência para o objeto *consumerPoPNo*, onde todas as políticas que possuírem o valor de *consumerPolicyByPoP* igual à *consumerPoPNo* representam políticas que foram recuperadas quando da ativação desta determinada PoP. Desta forma, a tabela *consumerPolicyTable* contém tanto políticas que foram carregadas individualmente quando políticas recuperadas em conjunto com PoPs.

O último grupo que será discutido neste trabalho é o *consumerEvents*. Os atributos *consumerEventNo*, *consumerEventStatus* e *consumerEventControl* possuem funções equivalentes aos objetos sinônimos existentes nas tabelas já discutidas. O atributo *consumerEventSender* referencia o elemento da rede que gerou o evento identifico pelo atributo *consumerEventID*. Nota-se que neste caso o objeto *consumerEventID* não é uma referência para o diretório LDAP, já que é indesejável e improvável que o elemento emissor do evento tenha acesso e compreenda as informações contidas no repositório do sistema. Este objeto é de fato o OID da *trap* SNMP enviada pelo emissor do evento. Assim, o PDP pode reagir de forma igual ou diferente nas situações onde um mesmo evento *(consumerEventID)* foi gerado por emissores diferentes *(consumerEventSender)*. Por fim, o objeto *consumerEventDate* é calculado automaticamente pelo PDP de forma a conter a data e hora do momento em que um evento foi recebido por ele. Como mencionado brevemente na seção anterior, esta tabela é configurada pelo serviço *snmptrapd* sempre que este receber uma *trap* enviada por algum elemento presente na rede. A rotina de avaliação do evento recebido e da decisão pela substituição ou não de uma política conforme o estado da PoP são

ativadas sempre que um novo evento é recebido pelo PDP através da configuração desta tabela.

Por fim, é importante mencionar que a disponibilidade de uma interface SNMP para controlar o elemento PDP pode ou não ser utilizada diretamente pelo gerente da rede. Ao mesmo tempo em que esta MIB está documentada e pode ser acessada através de comandos SNMP executados a partir de qualquer tipo de aplicação, o sistema Network Executive disponibiliza também uma interface gráfica acessível a partir de qualquer navegador Web, e que deve ser utilizada preferencialmente, pois é uma interface mais simples e mais amigável de ser utilizada do que comandos SNMP.

4.3. Esquema de Dados

Como já mencionado em seções anteriores, a arquitetura Network Executive prevê a utilização de um diretório LDAP como repositório das informações relevantes ao sistema, sejam elas cadastro de usuários, PDPs, PEPs, políticas, PoPs, eventos, entre outras. Estas informações são armazenadas através de estruturas de dados definidas através de um esquema, o qual é codificado conforme as convenções definidas pelo LDAP. Esta seção objetiva discutir este esquema de dados, senão na sua totalidade, pelo menos nas estruturas mais importantes para o funcionamento do sistema. As tabelas a seguir ilustram algumas destas estruturas que serão debatidas nesta seção.

Tabela 1. Objetos 'NetExecPolicy' e 'NetExecCos'

objeto	atributo	tipo	exemplo
NetExecPolicy	name	string	WebSite
	source-ip	string	*
	source-port	inteiro	*
	target-ip	string	192.168.0.100
	target-port	inteiro	80
	protocol	string	TCP
	dn-cos	dn	HTTPHighPriority
NetExecCoS	name	string	HTTPHighPriority
	throughput	inteiro	1000
	delay	inteiro	20
	jitter	inteiro	5
	loss-factor	inteiro	5

Tabela 2. Objetos 'NetExecPoP', 'NetExecPoP - FSM' e 'NetExecPoP - Script'

objeto	atributo	tipo	exemplo
NetExecPoP	name	string	IntranetSite
	description	string	PoP para rede Intranet
NetExecPoP-FSM	nodes	string	3
	edges	string	(1,2) (1,3)
	events	string	(delayIncrease) (networkOverload)
NetExecPoP-Script	pop-script	string	load WebSite

Tabela 3. Objeto 'NetExecEvent'

objeto	atributo	tipo	exemplo
NetExecEvent	name	string	delayIncrease
	trapOid	string	.1.2.3.4.5.6.7
	description	string	Delay increase detec.

Os objetos da tabela 1 são utilizados para armazenar políticas e classes de serviço *(Class of Service)*, respectivamente. Enquanto que no objeto *NetExecPolicy* estão definidas as

regras de identificação do fluxo a ser priorizado, no objeto *NetExecCoS* são relacionados os valores de QoS que devem ser aplicados em um determinado fluxo. A associação entre uma política e os valores de QoS que devem ser aplicados é feita através do atributo '*dn-cos*' do objeto *NetExecPolicy*. Observa-se que utilizando este esquema de dados é possível que políticas distintas utilizem os mesmos parâmetros de QoS sem a necessidade de especificar vários objetos *NetExecCoS* iguais.

O segundo conjunto de objetos é utilizado para definir uma PoP, incluindo sua máquina de estado finito, os eventos relacionados a cada transição e os *scripts* que devem ser executados quando da ativação de cada uma das possíveis transições. O objeto *NetExecPoP - FSM* pode ser considerado o mais importante deste grupo, pois nele estão contidas as informações que permitem ao PDP construir a máquina de estado associada a uma PoP. O número de nodos *(nodes)*, as transições *(edges)* e os eventos associados a cada transição *(events)* são informações armazenadas por este objeto. Já os *scripts* que são executados quando da ativação de uma transição são codificados no objeto *NetExecPoP - Script*. Nota-se que cada um destes objetos serve apenas para especificar um script associado a um nodo. Assim, haverá tantos objetos deste tipo quanto são os nodos presentes na máquina de estados.

Por fim, o objeto apresentado na tabela 3 é utilizado para especificar os eventos que o sistema Network Executive conhece e que, portanto, podem ser recebidos por qualquer PDP presente no sistema. O principal atributo deste objeto é o *trapOid*, o qual identifica o OID da *trap* SNMP que caracteriza este evento. Este valor de *trap* é utilizado para atualizar o objeto *consumerEventID* da MIB – POP, o qual é utilizado pelo sistema para decidir sobre a ativação ou não de uma determinada transição.

5. Conclusões e Trabalhos Futuros

Este artigo apresentou uma proposta de arquitetura capaz de prover um mecanismo para a substituição automática de políticas conforme a ocorrência de determinados eventos na rede. Além dos novos conceitos envolvidos e do modelo genérico capaz de prover este mecanismo, uma arquitetura real e implementada foi apresentada e discutida. Os elementos presentes nesta arquitetura, os protocolos utilizados por eles, bem como a MIB e o esquema de dados utilizados pelo sistema também foram abordados. O objetivo principal desta arquitetura é auxiliar o gerente na tarefa de manutenção da operação da sua rede quando da ocorrência de eventos que podem influenciar o comportamento da rede e, portanto, das aplicações e serviços existentes nela. Um sistema como o discutido neste artigo é importante, pois permite ao gerente se concentrar em outras tarefas referentes à operação e ao planejamento da sua rede, enquanto que um sistema autônomo monitora e configura a sua rede conforme o comportamento observado nela e o plano de ações previamente definido por ele. A utilização do formalismo introduzido pela PoP e sua máquina de estados finita permitem, inclusive, que este plano de ações seja expresso em uma linguagem mais formal e que serve também como dados de entrada para o sistema de gerenciamento.

Futuramente está prevista a construção de PDPs para outras tecnologias, já que hoje há apenas PDPs capazes de configurar um roteador Linux – CBQ. Desta forma, seria possível configurar uma gama maior de dispositivos e utilizar uma mesma arquitetura, no caso o sistema Network Executive, para controlar um número maior e mais heterogêneo de equipamentos. É importante ressaltar que a construção de novos

PDPs não resulta no aumento da complexidade de uso do sistema pelo gerente e, portanto, no gerenciamento da sua rede, uma vez que a interface gráfica se mantém inalterada.

Por fim, também está prevista a elaboração de um novo artigo contendo os dados e as conclusões obtidas através da utilização deste sistema em uma rede de testes. Esta validação já foi planejada e executada, mas os resultados obtidos não foram consolidados a tempo de serem incluídos neste artigo.

Referências

Boardman, B.; Saperia, J. (2002) "Are you a control freak ?", Network Computing magazine, janeiro de 2002.

Boardman, B.; Saperia, J. (2002A) "No standards, no policy, no management**",** Network Computing magazine, janeiro de 2002.

Coelho, G. (2001) "Uma análise de ferramentas para gerenciamento de redes baseado em políticas**",** Trabalho individual I, Instituto de Informática – UFRGS.

Coelho, G.; Granville, L.; Almeida, M.; Tarouco, L. (2001A) "Netwok Executive: A Policy-based Network Management Tool**",** IEEE Latin American Network Operations and Management Symposium (LANOMS 2001).

Conover, J. (1999) "Policy-based network management", Network Computing magazine, novembro de 1999.

Koch, T.; Krell, C.; Kramer, B. (1996) "Policy Definition Language for Automated Management of Distributed Systems", IEEE International Workshop on Systems Management.

Lupu, E.; Sloman, M. (1999) "Conflicts in Policy-based Distributed Systems Management", IEEE Transactions on Software Engineering.

Mahon, H.; Bernet, Y.; Herzog, S.; Schnizlein, J. (2000) "Requirements for a policy management system", Internet draft <draft-ietf-policy-req-02.txt>, The Internet Society.

Moore, B.; Ellesson, E.; Strassner, J.; Westerinen, A. (2001) "Policy core information model", Request for comments 3060, The Internet Society.

Moore, B. (2002) "Policy core information model extensions", Internet draft <draft-ietf-policy-pcim-ext-08.txt>, The Internet Society.

Ribeiro, M.; Granville, L.; Almeida, M.; Tarouco, L. (2001) "QoS Monitoring System on IP networks", IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS 2001).

Shepard, S. (2000) "Policy-based networks: hype and hope", IT Professional, Vol.2, No.1, pp.12-16.

Sloman, M. (1994) "Policy Driven Management for Distributed Systems", Journal on Networks and Systems Management, Vol. 2, no. 4, pp. 333-360.

Westerinen, A.; Schnizlein, J.; Strassner, J.; Scherling, M.; Quinn, B.; Perry, J.; Herzog, S.; Huynh, A.; Carlson, M.; Waldbusser, S. (2001) "Terminology for policy - based management", Request for comments 3198, The Internet Society.