

Solução baseada em políticas para Gerenciamento Integrado de QoS e Multicast em Redes IP, usando PCIME

Leandro Vaguetti¹, Lisandro Granville², Janilce Almeida², Liane Tarouco²

¹Faculdade Euro-Americana

Av. das Nações Sul, Trecho 0, Conjunto 5 – 70200-000 Brasília, DF

²Instituto de Informática – Universidade Federal do Rio Grande do Sul
Caixa Postal 15064 – 90501-970 Porto Alegre, RS

(vaguetti, granville, janilce, liane)@inf.ufrgs.br

Abstract. *Advanced applications, such as video conference, video on-demand, distance learning, among others, make use of computer networks as infrastructure support. However, in order to achieve an effective usage, the computer networks, on their turn, must offer some special capabilities to handle the applications needs. Quality of service (QoS) and multicast transmission support are some of the capabilities that must be supported. Besides the QoS and multicast support, it is necessary to supply a network management suitable to the applications expectations. Proposals that provide an individual solution for each of the capabilities have already been implemented. However, these solutions are not integrated, which makes the task of network management difficult and complex, due to the limitations inherent to non integrated systems. In this paper is showed a policy-based solution for integrated QoS and Multicast management using PCIME.*

Resumo. *Aplicações como videoconferência, vídeo sob-demanda, aplicações de ensino a distância, entre outras, utilizam-se das redes de computadores como infra-estrutura de apoio. Mas para que tal uso seja efetivo, as redes de computadores, por sua vez, devem fornecer algumas facilidades especiais para atender às necessidades dessas aplicações. Dentre as facilidades que devem ser fornecidas estão os suportes à qualidade de serviço (QoS - Quality of Service) e as transmissões multicast. Além do suporte a QoS e multicast nas redes, é necessário fornecer um gerenciamento da rede adequado às expectativas de tais aplicações. Soluções que fornecem gerenciamento de forma individual para tais facilidades, já foram propostas e implementadas. Entretanto, estas soluções não conseguem agir de modo integrado, o que torna a tarefa do gerente da rede extremamente complexa e difícil de ser executada, pois possibilitam um fornecimento não adequado das facilidades desejadas às aplicações. Neste artigo é apresentada uma solução baseada em políticas para gerenciamento integrado de QoS e multicast, usando PCIME.*

1. Introdução

As novas redes de computadores, como a Internet2 e a RNP2, viabilizam a utilização de aplicações avançadas como videoconferência, vídeo interativo, bibliotecas digitais e laboratórios virtuais. Tais aplicações exigem da rede utilizada um conjunto mais amplo de serviços, com complexidade superior ao serviço de melhor esforço (*best-effort*) oferecido atualmente. Tal conjunto de serviços deve cuidar para que o tráfego gerado mantenha-se de acordo com os níveis de qualidade esperados

pelas aplicações. A inserção da noção de qualidade de serviço (QoS - *Quality of Service*) é extremamente importante nos ambientes de rede para que as aplicações possam operar com um nível de garantia mínimo, ausente em redes "convencionais" como as redes baseadas no protocolo IP [McDysan, 1999]. A capacidade de garantir o nível de qualidade da rede é um dos requisitos que devem ser encontrados nas novas redes de computadores.

Além de QoS, uma capacidade comumente requisitada pelas novas aplicações é a capacidade de transmissão *multicast*. O multicast tem a função de endereçar um grupo de um ou mais dispositivos de rede, oferecendo vantagens para aplicações que necessitem de otimização quanto ao uso da rede e utilizem-se de grupos multicast com diversos participantes recebendo dados simultaneamente. As aplicações mais críticas normalmente necessitam trocar informações com várias estações simultaneamente, além de esperar um comportamento adequado da rede de acordo com suas necessidades (e.g. largura mínima de banda garantida). Portanto, a disponibilidade de QoS e multicast nas redes apresenta-se como uma facilidade realmente necessária das novas aplicações.

Supondo que as facilidades necessárias para que as aplicações executem da forma esperada pelos usuários sejam oferecidas pela rede, espera-se que as aplicações possam operar adequadamente. Entretanto, podem ocorrer eventos na rede que acabem por degradar o serviço oferecido, inicialmente e, por consequência, diminuir o desempenho da aplicação. Torna-se, assim, necessário o uso de um mecanismo de gerenciamento que possa identificar e controlar os eventos ocorridos, assim como, de que forma as facilidades utilizadas estão sendo disponibilizadas às aplicações.

O gerenciamento de QoS e multicast atualmente pode ser realizado através de várias ferramentas, porém a QoS é gerenciada através de ferramentas específicas para o gerenciamento de QoS, levando em conta apenas aspectos relacionados à QoS, de forma única. Podem-se citar alguns esforços para facilitar o gerenciamento de QoS: IntServ-MIB, RSVP-MIB, DiffServ-MIB, etc. Da mesma forma, uma rede multicast é gerenciada através de ferramentas específicas ao multicast: IPMroute-MIB, IGMP-MIB e PIM-MIB. Entretanto, as aplicações avançadas necessitam das duas facilidades de forma conjunta, e conseqüentemente necessitam de um mecanismo de gerenciamento que opere também de forma única, o que não ocorre nas ferramentas de gerenciamento atuais, onde o gerenciamento de QoS e multicast é tratado de forma não integrada.

O principal objetivo deste trabalho é apresentar uma solução de gerenciamento integrado para QoS e multicast em redes IP. Optou-se pelo uso do gerenciamento baseado em políticas (PBNM - *Policy-based Network Management*) [Sloman, 1994] como elemento facilitador da integração. Nesse contexto, é necessário definir políticas para gerenciamento integrado de QoS e multicast, onde a definição das políticas seguem as recomendações do *Policy Group* do IETF (*Internet Engineering Task Force*). Além disso, o IETF propõe uma arquitetura geral de gerenciamento baseado em políticas, definindo quem são e como operam cada um de seus elementos (e.g. PDP's e PEP's) [Westerinen et al., 2001], porém, baseado nas definições estudadas, torna-se necessário acrescentar novos elementos à arquitetura inicial para que as políticas definidas por esse trabalho sejam aplicadas adequadamente (e.g. monitores que sinalizem eventos ocorridos na rede). Num segundo momento é apresentado o protótipo implementado, baseado na arquitetura definida anteriormente. São comentados também os testes realizados para validar a solução proposta e o protótipo.

Este trabalho está organizado da seguinte forma: a seção 2 apresenta os trabalhos relacionados ao gerenciamento de QoS e multicast. A seção 3 apresenta uma proposta de integração do gerenciamento dos serviços de QoS e multicast usando gerenciamento baseado em

políticas. Na seção 4 é apresentada a implementação e testes, baseados nas definições propostas na seção 3. Por fim, o seção 6 encerra este trabalho apresentando as conclusões.

2. Trabalhos Relacionados

Em se tratando de gerenciamento de QoS e multicast, diversas soluções podem ser encontradas, dentre elas estão MIB's definidas, tanto para gerenciamento de QoS quanto para gerenciamento de multicast, algumas ferramentas, e ambientes desenvolvidos para dar suporte a tais gerenciamentos e alguns esforços para padronizar tecnologias de gerenciamento emergentes. Nesta seção serão apresentadas as soluções encontradas atualmente para o fornecimento de gerenciamento de QoS e multicast.

O gerenciamento de QoS pode ser alcançado através de várias ferramentas já existentes no mercado, ferramentas estas que normalmente possuem sua implementação baseada no padrão de gerenciamento Internet (SNMP- *Simple Network Management Protocol*) e que podem adicionar ao seu conjunto de objetos gerenciáveis os objetos gerenciáveis encontrados nas MIB's definidas, para padronizar o gerenciamento de arquiteturas para fornecimento de QoS como Serviços integrados (IntServ) e Serviços diferenciados (DiffServ). Além disso, alguns protocolos definidos para possibilitar a implementação de tais arquiteturas, como é o caso do RSVP, também podem ser gerenciados através de MIB's. O gerenciamento de QoS também pode ser alcançado através de ambientes que não se utilizam apenas de SNMP para prover gerenciamento. Nestes ambientes, como é o caso do ambiente de gerenciamento de QoS QAME [Granville and Tarouco, 2001], utilizam-se de formas alternativas (e.g PBNM) para fornecer o gerenciamento necessário.

Além das ferramentas e tecnologias apresentadas anteriormente, existe uma que tem se destacado como tendência no gerenciamento de *QoS*. Trata-se do gerenciamento baseado em políticas (PBNM), tecnologia de gerenciamento emergente que visa fornecer ao gerente facilidades de gerenciamento, através da definição de políticas de alto nível na rede. Dois modelos de informação têm sido discutidos no IETF, com relação ao gerenciamento de QoS. Um deles tornou-se recentemente uma RFC e o outro se encontra em forma de *draft*, e propõem mecanismos para definição de políticas para gerenciamento de QoS, sob dois focos distintos. O PQIM (*Policy QoS Information Model*) [Snir et al., 2003] propõe a definição de políticas mais abstratas, independentes da tecnologia utilizada nos dispositivos aos quais as políticas serão aplicadas, preocupando-se somente em armazenar os parâmetros que ditam o comportamento esperado da rede, com relação à QoS exigida pelas aplicações. Neste modelo as políticas podem ser definidas para aplicação em diversos dispositivos ou domínios administrativos, sem modificação da política. O outro foco abordado pelo IETF é o armazenamento de políticas orientada a um dispositivo específico, definindo a QoS de acordo com a tecnologia oferecida por cada dispositivo. Esta proposta é chamada pelo IETF de *Information Model for describing Network Device QoS Datapath Mechanisms* [Moore et al., 2003].

Quanto ao gerenciamento de multicast, com o surgimento de infra-estruturas de serviços e protocolos de multidifusão, tornou-se necessária a disponibilização de recursos de apoio à gerência dos mesmos. Dentre os recursos disponibilizados estão principalmente MIB's, que abrangem de aspectos gerais até aspectos específicos dos protocolos utilizados em transmissões multicast na internet. O gerenciamento de multicast também pode ser feito através de ferramentas que utilizam ou não as MIB's, definidas para esta tarefa. Em se tratando de MIB's, uma das mais importantes para o gerenciamento multicast é a MIB de roteamento multicast (também denominada IPMRRoute-MIB). Esta é proposta na RFC 2932, tendo como objetivo definir objetos gerenciáveis que auxiliem na gerência do roteamento multicast de forma genérica, onde os objetos gerenciáveis dependentes de algum protocolo de roteamento multicast devem ser definidos nas MIBs específicas a cada

protocolo.

Para tornar o gerenciamento de roteamento multicast completo, foram definidas MIB's que oferecem objetos gerenciáveis para cada protocolo de roteamento multicast de forma específica, visando fornecer o gerenciamento de aspectos relacionados apenas a um determinado protocolo. São encontradas MIB's para os protocolos de roteamento multicast DVMRP e PIM . O protocolo que controla os grupos multicast (IGMP), que também possui uma MIB definida para fornecer seu gerenciamento. A *Internet Group Management Protocol* MIB (IGMP-MIB) define objetos gerenciados, que permitem monitorar a operação do protocolo IGMP em *hosts* e roteadores.

3. Gerenciamento Integrado de QoS e Multicast usando PCIME

Esta seção apresenta o gerenciamento baseado em políticas (PBNM -*Policy-Based Network Management*) como estratégia de gerenciamento adotada para desenvolver a solução proposta neste trabalho. Em seguida, serão apresentados os problemas que podem ser percebidos quando da utilização de PBNM, em sua forma tradicional, para desenvolver tal solução. Por isso, a implementação do trabalho requer adaptações no modelo de PBNM tradicional. Assim, as modificações sugeridas no modelo atual, visando ao desenvolvimento da solução, também são descritas. Por fim, esta seção apresenta a solução proposta para fornecimento de gerenciamento integrado de QoS e multicast.

De acordo com as definições apresentadas pelo IETF, dois modelos de organização de políticas com baixo nível de abstração podem ser usados. O primeiro modelo, conhecido como *Policy Core Information Model - PCIM* [Moore et al., 2001], é o modelo tradicional para organização de políticas, e tem sido utilizado como referência atualmente. O segundo modelo, chamado *Policy Core Information Model Extensions - PCIME* [Moore, 2003], estende o modelo tradicional, adicionando novas classes e atributos, tornando a organização de políticas, através deste modelo, mais detalhada e flexível. O novo modelo foi padronizado, recentemente, e ainda não possui muitas implementações disponíveis, apesar de seu potencial com relação à organização de políticas.

O modelo de dados representado no PCIM possibilita a definição de políticas limitadas com relação à QoS e multicast, pois cada política pode ser composta apenas por um conjunto de ações e condições, um conjunto de regras (que por sua vez são compostas de ações e condições) e um conjunto de grupos de regras. Tais regras são definidas linearmente, o que acaba limitando a construção das políticas com relação ao aspecto hierárquico das mesmas. Por outro lado, o único mecanismo para definir o esquema de execução de uma política é um atributo de prioridade, vinculado às regras, de modo a determinar qual a prioridade de execução entre regras de uma mesma política. Assim, como comentado anteriormente, políticas para QoS e multicast têm uma característica especial de execução devido a dinâmica de localização dos fluxos, fator que acaba tornando inviável a definição de tais políticas, utilizando o PCIM como modelo. Outra alternativa é a adoção de um mecanismo auxiliar para definição políticas em conjunto com o PCIM. Um exemplo de política para QoS e multicast, seguindo unicamente o PCIM, é apresentada na Figura 1.

A política apresentada na Figura 1 é aplicada da seguinte maneira. No momento especificado nas políticas, as seguintes regras são aplicadas: `EnableMulticastSupport` e `QoSforMulticastGroup`. Quando a regra `QoSforMulticastGroup` é aplicada, os recursos necessários para QoS são alocados. As configurações necessárias para tráfego de fluxo multicast são realizadas no momento em que a regra `EnableMulticastSupport` também é aplicada. No momento em que um novo membro entra no grupo multicast, os recursos necessários com relação

```

Policy: Policy for QoS and multicast
Rule: EnableMulticastSupport
    if (timeOfDay >= 4pm) and (timeOfDay <= 6pm)
    then
        MroutedEnabled = true
        QueryInterval = 1000 ms
        MRoutingProtocol = DVMRP
Rule: QoSforMulticastGroup
    if (timeOfDay >= 4pm) and (timeOfDay <= 6pm)
    and (DestAddr == 224.0.0.214)
    then
        Bandwidth = 300 Kbps
        MaxJitter = 10%
        MaxLoss = 30%

```

Figure 1: Política para gerenciamento de QoS e multicast usando PCIM

à QoS já estão alocados. Quando qualquer membro do grupo multicast deixa de fazer parte do grupo multicast, os recursos relativos à QoS continuam alocados desnecessariamente.

Em qualquer uma das arquiteturas de fornecimento de QoS, onde políticas como a apresentada na Figura 1 são aplicadas para fornecimento de gerenciamento de QoS e multicast, o desperdício de recurso é inaceitável, devido à utilização desnecessária de memória dos dispositivos intermediários, e principalmente se disciplinas de enfileiramento de pacotes como *Custom Queued* estiverem sendo utilizadas.

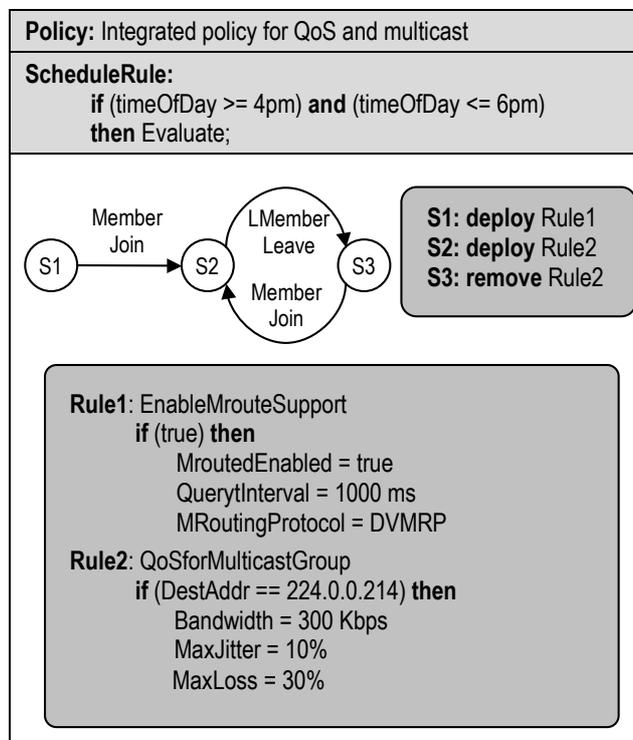


Figure 2: Política para gerenciamento integrado de QoS e multicast usando PCIM e FSM

Neste contexto, considerando a forma ideal de aplicação de uma política para QoS e multicast de forma integrada, o modelo sugerido no PCIM não suporta a definição de políticas

que expressem o comportamento esperado da rede nessas situações. Não existem no PCIM mecanismos que possibilitem a definição da estratégia de execução das regras de forma dinâmica. Sendo assim, é necessário agregar elementos externos à política para determinar as estratégias de execução. Uma das formas de definição desta estratégia é utilizar FSM (*Finite State Machine*) [Vaguetti et al., 2003] [Granville et al., 2003]. Utilizando-se FSM como recurso auxiliar, a política é definida como apresentado na Figura 2.

No exemplo apresentado na Figura 2 tem-se uma política que deve ser aplicada dentro de um determinado período de tempo (no caso, entre 4 e 6 horas da tarde). Abstraindo os aspectos temporais da política, existem algumas regras que devem ser aplicadas em resposta a alguns eventos reportados pela rede. Tais regras são disparadas através de *scripts*, que por sua vez estão associados aos estados da FSM. A transição dos estados da regra está diretamente ligada aos eventos. No exemplo, tem-se 2 regras, uma que configura aspectos relativos à multicast (*Rule1*) e outra que configura os aspectos relativos à QoS (*Rule2*). Além disso, a política possui três *scripts*, onde S1 aplica a *Rule1*, S2 aplica a *Rule2* e S3 remove a *Rule2*. De acordo com a FSM definida, quando a política é aplicada, o estado 1 é disparado (nesse caso, o *script* S1 é executado) configurando o multicast da rede, se o evento correspondente à entrada de um novo membro no grupo especificado, é percebido, então, a FSM passa ao estado 2 (S2) onde a QoS é configurada. Se é percebida a saída de todos os membros do grupo, então o estado 3 (S3) é executado, liberando os recursos de QoS. Entretanto, se algum membro voltar a fazer parte do grupo, os recursos de QoS devem ser novamente configurados, nesse caso, a FSM volta ao estado 2 (S2). Quando a política não for mais válida, temporalmente todas as regras são removidas.

Além do PCIM, existe outro modelo proposto, recentemente, para organizar a definição de políticas. Utilizando-se do modelo apresentado no PCIME, é possível definir políticas integradas para QoS e multicast, sem a necessidade de uso de um mecanismo auxiliar para definir a estratégia de execução das regras em uma política. A principal diferença entre PCIME e PCIM, com relação à estratégia de execução, é a criação de uma classe intermediária chamada `PolicySet`, que possui relacionamentos com outras classes que permitem as regras serem definidas com um número ilimitado de níveis hierárquicos. Além disso, a mesma classe ainda define um atributo (`PolicyDecisionStrategy`) usado para definir a estratégia de prioridade dentre um conjunto de regras organizadas de forma hierárquica. Outra modificação importante foi a inclusão de um novo atributo (`ExecutionStrategy`) na classe `PolicyRule`. Tal atributo permite definir qual a estratégia de execução de cada regra. Através desse, é possível configurar se uma regra deve ser aplicada durante todo o tempo de execução da regra, se deve ser aplicada até o momento que falhar a primeira vez, ou se deve ser aplicada sempre que tiver sucesso de execução. Uma política, com o mesmo objetivo de aplicação dos modelos anteriores, pode ser definida usando o novo modelo, como apresentado na Figura 3).

A forma como a política definida, usando PCIME, é aplicada, é descrita a seguir. No momento em que a política é aplicada a `Rule1` é verificada, nesse caso, é verificada a condição de tempo de aplicação da regra. Também na `Rule1` são definidos dois atributos. O atributo `ExecutionStrategy` é setado de forma que a regra seja verificada durante todo o tempo especificado como condição; já o atributo `DecisionStrategy` define que todas as ações componentes da regra devem ser executadas conjuntamente. Num nível hierárquico inferior está a `Rule2` (que faz parte da lista de ações da `Rule1`). Tal regra habilita o suporte à multicast, determina que tais valores devem ser mantidos até que a regra imediatamente superior (`Rule1`) deixe de executar. Além disso, a `Rule2` possui duas ações que disparam duas novas regras (`Rule3` e `Rule4`) e os atributos `DecisionStrategy` e `ExecutionStrategy`. No caso das regras `Rule3` e `Rule4`, uma é contrária à outra, ou seja, se o fluxo multicast para o grupo 224.0.0.13 for percebido, a `Rule3` torna-se válida e suas ações (configurações relativas à QoS)

```

Policy: Policy for QoS and multicast
Rule1: TimeDefinition
    if (pcimTPCtimeOfDay >= 8am) and (pcimTPCtimeOfDay <= 11am)
    then
        Rule2: EnableMulticastSupport
        if (pmcastMroutedEnabled == false) then
            pmcastMroutedEnabled = true
            pcimExecutionStrategy = 2

            Rule3: QoS Multicast Group Prune
            if (pmcastGroupIPv4 == 224.0.0.13)
            then
                qpBandwidthUnits = 0
                qpMinBandwidth = 300 Kbps
                qpMaxBandwidth = 450 Kbps
                pcimDecisionStrategy = 2
                pcimExecutionStrategy = 2

                Rule4: QoS for Multicast Group Leave
                if (pmcastGroupIPv4 != 224.0.0.13)
                then
                    clear configuration in Rule3
                    pcimExecutionStrategy = 2
                    pcimDecisionStrategy = 2
                pcimExecutionStrategy = 2
                pcimDecisionStrategy = 2

```

Figure 3: Política para gerenciamento integrado de QoS e Multicast usando PCIME

são executadas; caso contrário, se a fluxo multicast, em questão, não for percebido, a Rule4 torna-se válida, retirando as configurações setadas pela Rule3. O elemento que torna possível esse tipo de execução é o atributo `ExecutionStrategy`, que define que as duas regras devem sempre ser aplicadas, independentemente de falharem ou não, enquanto as regras superiores forem válidas.

Nesta seção foram apresentados os problemas encontrados quando da definição de políticas para gerenciamento integrado de QoS e multicast, usando o modelo tradicional de organização e as formas possíveis de solucionar tais problemas, tendo como base as orientações do IETF. Entretanto, nas duas formas de organização possíveis, modificações na arquitetura tradicional de PBNM são necessárias. No caso da adoção de PCIM como modelo, a arquitetura deve ser complementada com um mecanismo, como por exemplo FSM, para definir a estratégia de execução. Utilizando o PCIME, não é necessário um mecanismo auxiliar, entretanto, elementos que verifiquem constantemente a validade das condições das regras, devem existir para dar suporte ao que é definido no atributo `ExecutionStrategy`. A forma de organização de políticas adotada, nesse trabalho, e a nova arquitetura de PBNM proposta para suportar tal modelo de organização, serão discutidas à seguir.

Duas formas de organização de políticas foram apresentadas anteriormente: uma forma de organização usando o modelo PCIM e outra usando o PCIME. Neste trabalho, a forma de organização de políticas adotada é o PCIME. Tal modelo foi escolhido, pois, além de herdar todas as facilidades de organização de políticas contidas em PCIM (e.g `PolicyCondition` e `PolicyAction`), ainda oferece elementos que possibilitam a declaração de políticas, sem que seja necessário acrescentar novos elementos na estrutura básica de organização das políticas para determinar seu funcionamento.

Dentre os novos atributos oferecidos em PCIME, estão atributos que possibilitam

a definição da forma como as regras devem ser aplicadas internamente à política (e.g. `pcimDecisionStrategy` e `pcimExecutionStrategy`), e um conjunto básico de classes e atributos que definem variáveis e valores (e.g. `pcimVariable` e `pcimValue`), que podem ser utilizados na definição das condições e ações, de uma política individualmente. Entretanto, as classes e atributos oferecidos inicialmente em PCIME atendem apenas às necessidades de definição de políticas básicas (e.g. políticas para controle de fluxo e políticas de filtragem de pacotes). Assim, se políticas mais específicas a uma determinada tecnologia necessitam ser definidas, torna-se necessária a criação de atributos e classes específicos para declaração de tais políticas (e.g. atributos e classes para definição de variáveis e valores relacionados à QoS e multicast). Porém, PCIME já possui suporte à declaração de novas classes e atributos, permitindo que novas variáveis e valores sejam agregados à PCIME, de forma a permitir a declaração de condições e ações não-previstas anteriormente. A modelagem apresentada em PCIME permite que a organização dos dados torne-se extensível (e.g. `pmcastGroupIPv4VariableAuxClass`, classe definida de forma a estender o PCIME para que variáveis que definam endereços de grupos multicast possam ser declarados em políticas de multicast).

Como comentado anteriormente, além de PCIME, outros modelos de informação são necessários para declaração de políticas integradas de QoS e multicast. Assim, a declaração de modelos auxiliares é necessária, pois atributos diretamente relacionados à QoS e multicast não são encontrados diretamente em PCIM e PCIME. Entretanto, tais políticas não podem ser declaradas, sem a utilização de tais elementos (e.g. `qpMaxDelay`, variável definida em um modelo de informação auxiliar para que o atraso máximo de um pacote durante uma troca de mensagens pudesse ser definida em políticas para QoS). Para declarar os atributos relativos à QoS, este trabalho utiliza-se de um modelo de informação também definido pelo IETF: trata-se do PQIM (*Policy QoS Information Model*). No caso da declaração dos atributos relativos à multicast, torna-se necessária a criação de um modelo auxiliar que suporte a declaração de variáveis e valores específicos para multicast, pois o IETF ainda não oferece nenhum tipo de suporte à declaração de tais políticas.

A Figura 3 apresenta um exemplo de política integrada para QoS e multicast, utilizando-se do PCIME e alguns modelos auxiliares que permitem a definição de tal política. A política apresenta um conjunto de regras que permitem o gerenciamento de um fluxo multicast que necessita de garantias de QoS. A forma como as regras estão dispostas internamente à política, e as condições e ações utilizadas em tais regras determinam como a política vai ser aplicada no decorrer do tempo válido para funcionamento da mesma. A política exemplo possui quatro regras, sendo que algumas destas regras estão declaradas internamente a uma regra posicionada de forma superior na hierarquia de regras. A principal regra à política (`Rule1`) define o período de tempo de aplicação da regra como condição, e dentre suas ações, é declarada uma nova regra (`Rule2`), que controla o suporte a fluxos multicast. Dentre as ações da `Rule2` estão definidas duas novas regras (`Rule3` e `Rule4`) que, de acordo com um determinado fluxo multicast, atendem ou não às necessidades de QoS exigidas. Na declaração da política é possível também observar a utilização de atributos definidos em PCIME (e.g. `pcimDecisionStrategy`), PQIM (e.g. `qpBandwidthUnits` e variáveis e atributos definidos neste trabalho para declaração de condições e ações relacionadas à multicast (e.g. `pmcastMRouteEnable` e `pmcastGroupIPv4`). Uma apresentação mais detalhada da política é feita no final da seção.

Além dos atributos citados anteriormente, um importante atributo, definido no PCIME e utilizado na política exemplo, é o atributo `ExecutionStrategy`. Tal atributo exige da arquitetura tradicional de PBNM um controle que não seria necessário utilizando PCIM, pelo menos de forma explícita. Entretanto, para que a política exemplo ou qualquer política definida, baseada na organização fornecida no PCIME, seja aplicada corretamente em uma arquitetura de PBNM,

é necessário que a arquitetura tenha suporte a esse tipo de controle. A arquitetura tradicional de PBNM, definida antes da padronização do PCIME, não fornece alguma forma de controle de como as políticas estão sendo executadas. Após a definição do PCIME, algum elemento na arquitetura deve ser responsável por desempenhar essa função vital, sob pena de que políticas que se utilizem de PCIME não possam ser aplicadas. O elemento mais adequado para desenvolver este tipo trabalho na arquitetura tradicional seria o PDP, porém, segundo literatura estudada, a função do PDP é de tradução de políticas de alto nível para políticas mais específicas, tornando-as possíveis de serem aplicadas nos PEPs. Uma nova arquitetura foi definida neste trabalho. Tal arquitetura busca fornecer o suporte necessário para utilização de políticas baseadas no PCIME, e apresenta com maior nível de detalhes os elementos necessários para aplicação de políticas integradas de QoS e multicast utilizando-se do PCIME (Figura 4).

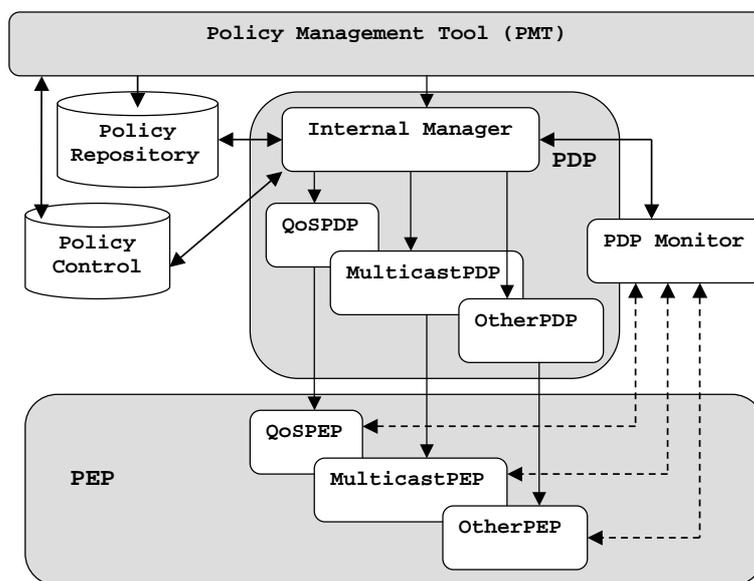


Figure 4: Novos elementos na Arquitetura PBNM

A Figura 4 apresenta a nova arquitetura e um conjunto de novos elementos adicionados à arquitetura tradicional, com objetivo de prover suporte ao PCIME. Os elementos PMT, PDP, PEP e PR já são tradicionais, porém novos elementos são destacados internamente aos elementos anteriores, e estes serão comentados a seguir.

Inicialmente são divididas as funções do PDP em duas funcionalidades diferentes. A primeira funcionalidade é a de controle de aplicação das políticas. Tal funcionalidade controla a execução de cada regra e onde cada política vai ser aplicada no ambiente de gerenciamento. Seus elementos principais são o **Policy Control**, que armazena informações que ditam em quais PDPs e PEPs determinada política deve ser aplicada, além do status de aplicação de cada política, com relação aos elementos (e.g. PEPs). Tal elemento é atualizado quando necessário pelo **Internal Manager**. **Internal Manager** é o elemento que controla a execução de uma política e conseqüentemente cada regra dentro da política. Uma de suas funções é verificar a validade de cada política com relação a cada PEP associado, e atualizar o **Policy Control** para que, a partir destas informações, o **PMT** possa executar as ações necessárias com relação à política. Outra função do **Internal Manager** é verificar questões temporais relacionadas à política, como por exemplo, se uma política deve ser aplicada entre quatro e cinco horas da manhã. Além das funções anteriores, o **Internal Manager** ainda é responsável por verificar a relação hierárquica de execução de uma política (e.g. quando as regras filhas tornam-se inválidas, a regra pai deve tornar-se inválida também, e vice-versa) e verificar o status de execução das regras

dentro da política, de acordo com o que está definido no atributo `ExecutionStrategy`. Para que o `Internal Manager` possa fazer a verificação do status de execução das regras, um outro elemento definido é utilizado para auxiliá-lo. Trata-se do `PDPMonitor`.

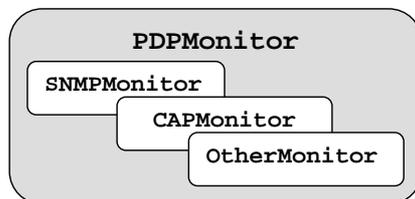


Figure 5: PDPMonitor

O `PDPMonitor` pode ser considerado uma extensão do PDP ou um novo elemento na arquitetura. Neste trabalho ele é considerado uma extensão do PDP, pois sua função é dar suporte às decisões do `Internal Manager`, que está situado internamente ao PDP. O `PDPMonitor` tem como função principal verificar o *status* das condições definidas em cada regra (e.g. verificar se a existência de fluxo multicast para o grupo 224.0.0.13, de acordo com a `Rule3` da política exemplo). Tais condições devem ser notificadas ao `PDPMonitor` pelo `Internal Manager` que, por sua vez, verifica as condições atribuídas para verificação. Diversas verificações de condições podem ser solicitadas ao `PDPMonitor`. Por isso é necessário que ele suporte várias formas de monitoração (e.g. através de snmp ou através de captura de pacotes). A Figura 5 apresenta o `PDPMonitor` com um nível maior de detalhe, destacando-se alguns tipos de monitoração que podem ser suportadas pelo elemento (e.g. `SNMPMonitor` e `CAPMonitor`).

A segunda funcionalidade do PDP é a tradução das políticas definidas num nível mais abstrato para políticas de mais baixo nível, objetivando a aplicação direta a um PEP do sistema. Neste contexto, a arquitetura utiliza-se de elementos tradutores específicos para cada tecnologia aplicada (e.g. `QoS PDP` e `Multicast PDP`). Tais elementos são acionados pelo `Internal Manager` no momento que as regras relacionadas a cada tecnologia tornam-se válidas. Por exemplo, quando um fluxo multicast para o grupo 224.0.0.13 é percebido, as ações definidas na `Rule3` são aplicadas. Para facilitar a aplicação, PEPs específicos para cada tecnologia também são propostos (e.g. `QoS PEP` e `Multicast PEP`). No exemplo citado anteriormente, os elementos internos `QoS PDP` e `QoS PEP` são utilizados para aplicação da regra; em outro contexto, poderiam ser usados outros elementos tradutores e aplicadores.

A seguir, será descrito como se comporta a aplicação da política apresentada, na Figura 3, na nova arquitetura apresentada, na Figura 4. Inicialmente a política é definida e armazenada no `Policy Repository`, neste momento a definição de quais PDPs e PEPs aplicarão a política, também é armazenada no `Policy Control`. Todas estas definições são feitas através do `PMT`, que também tem a função de notificar ao PDP que uma nova política está armazenada no PR. Quando tal notificação acontece, o PDP, através do `Internal Manager`, baixa as políticas para seu módulo de controle e inicia o processo de controle das regras da política. Como as regras das políticas definidas, usando `PCIME` podem ter vários níveis hierárquicos, o `Internal Manager` preocupa-se também com a hierarquia de execução. A primeira regra a ser verificada na política exemplo é a `Rule1`. Esta regra apresenta uma condição temporal, definindo que tal regra e suas ações devem ser aplicadas entre oito e onze horas da manhã. No momento que esta regra se torna válida, suas ações são aplicadas, neste caso, é disparada a `Rule2`. Os atributos `pcimExecutionStrategy` e `pcimDecisionStrategy` também são setados, garantindo que regra deve ser verificada, até que a condição de tempo seja inválida.

Num segundo momento, a `Rule2` é verificada, neste caso, o elemento `PDPMonitor` é

utilizado para verificar se a capacidade de rotear fluxo multicast é válida nos PEPs em questão. Em caso negativo, as ações determinadas na regra devem ser aplicadas. Dentre as ações, destaca-se a habilitação do roteamento multicast, através do atributo `pmcastMRoutedEnabled` (utilizando os elementos `PDPMulticast` e `PEPMulticast`) e duas novas regras: `Rule3` e `Rule4`. A verificação da validade dessa regra é determinada para que deixe de ser verificada somente quando a `Rule1` torna-se inválida. As regras `Rule3` e `pmRule4` passam a ser verificadas utilizando novamente o `PDPMonitor`. Neste caso as condições das regras são contrárias, uma verifica se existe fluxo multicast para o grupo `224.0.0.13`, e a outra verifica se não existe tal fluxo. Se o fluxo existir, a `Rule3` é aplicada configurando-se os parâmetros relativos à QoS usando os elementos `PDPQoS` e `PEPQoS`; caso contrário, a `Rule4` torna-se ativa, desconfigurando o QoS aplicado pela `Rule3`. As duas regras são verificadas até que a regra imediatamente superior (`Rule2`) torne-se inválida, porém as duas regras são constantemente verificadas, devido à utilização do atributo `pcimExecutionStrategy`, na definição da política.

4. Implementação e Testes

O protótipo implementado está integrado ao ambiente de gerenciamento QAME (*QoS-Aware Management Environment*), e é a partir do ambiente gráfico oferecido pelo sistema, que serão demonstradas algumas operações e o funcionamento do protótipo.

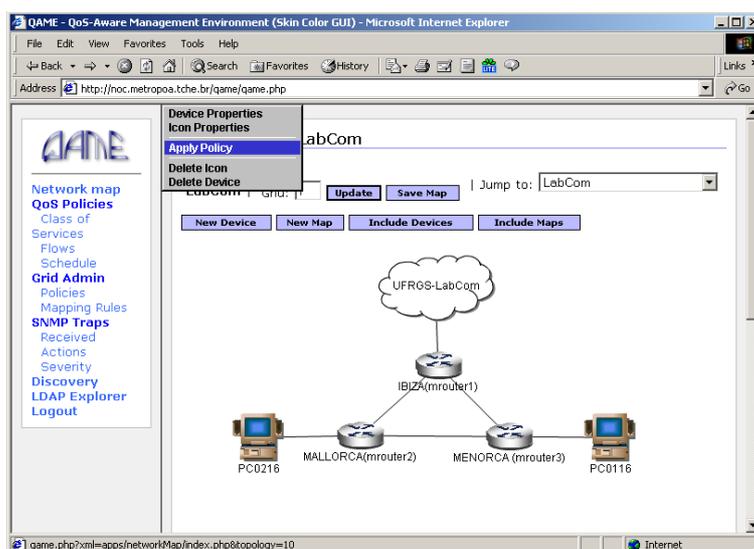


Figure 6: Módulo de associação de políticas e PDPs aos PEDs

Os primeiros módulos apresentam o funcionamento da operação mais executada por parte do administrador da rede, que se resume ao fato de atribuir políticas previamente cadastradas aos PEPs, nos quais as políticas serão aplicadas. Para que as políticas sejam aplicadas, o sistema deve possuir informações prévias das próprias políticas, dos PEPs, onde as políticas serão aplicadas, e dos PDPs que irão aplicar tais políticas no momento oportuno. Os módulos que tratam do cadastro de políticas, PEPs e PDPs, também foram implementados, mas não serão apresentados no artigo.

Inicialmente é possível determinar, através do ambiente gráfico, quais políticas são aplicadas a cada PEP e quais PEPs vão ser associados a cada PDP existente no sistema. Para que isso seja possível, é necessário selecionar o elemento dentro de uma determinada topologia, abrir um pequeno menu associado a cada elemento, através de uma combinação de teclas, e selecionar a opção "Apply Policy" (Figura 6). O sistema identificará se tal elemento foi registrado como

PEP ou PDP no sistema. Se o elemento foi cadastrado como PEP, uma tela, com a opção de associar políticas ao elemento selecionado, será apresentada ao usuário. Através desse módulo de associação, várias políticas podem ser associadas ao elemento selecionado. Ainda é possível definir um nome, uma breve descrição, além de editar e deletar tal associação, se for necessário.

Figure 7: Módulo de controle de Regras

A Figura 7 demonstra o módulo que permite que regras sejam criadas. Tais regras podem possuir propriedades, condições e ações. A definição de como as regras podem ser declaradas está descrita no PCIME. E para suportar tais definições, o protótipo ainda permite que regras sejam associadas a ações de uma regra de nível superior. Neste contexto, é permitida a criação de uma hierarquia de regras. Após declaradas as regras uma política pode ser finalmente criada. É importante salientar que uma política só pode ser criada se uma regra existir, ou seja, todas as políticas devem possuir ao menos uma regra associada.

Além do módulo web, apresentado acima, outros elementos da arquitetura (Figura 4) foram implementados. O elemento denominado PMT, é um software baseado na web, que permite que políticas e seus devidos controles sejam adicionados ao sistema. O PMT foi desenvolvido com tecnologias com PHP4, HTML e XML. As políticas declaradas através do PMT são armazenadas em um serviço de diretórios LDAP (*LDAP Directory Services*), de acordo com o modelo definido no PCIM e PCIME.

Além das políticas, a parte de controle é definida através do PMT e também armazenada no serviço de diretórios LDAP. A base de controle armazena informações de relacionamento entre políticas que devem ser aplicadas nos PEPs e os PDPs que deverão efetivamente aplicar tais políticas, além de informação de *status* de aplicação das políticas.

Uma vez declarada, a política e seus relacionamentos de controle se tornam disponíveis ao PDP. Tal controle é feito definindo elementos no serviço de diretórios LDAP, que indicam que uma nova política deve ser aplicada por determinado PDP do sistema, sendo função dos

PDPs verificarem constantemente a existência de novas políticas a serem baixadas e aplicadas. O elemento interno ao PDP, chamado `Internal Manager`, tem como uma de suas funções verificar a existência de novas políticas e baixá-las quando elas forem disponibilizadas. Tais políticas são buscadas no `Policy Repository`. A partir desse momento, o `Internal Manager` passa a controlar as questões temporais de aplicação das políticas e cada regra vinculada a política, se houverem. As políticas são armazenadas internamente ao `Internal Manager`, em forma de objetos hierárquicos, que são capazes de representar toda a hierarquia de objetos proposta em CIM, PCIM e PCIME, assim como partes de alguns modelos específicos (e.g. PQIM). O elemento `Internal Manager` é responsável por outras funções internas ao PDP.

Para que o controle de execução seja efetuado, é necessário que existam elementos na rede que verifiquem as condições impostas através das regras na política. O papel de verificação e validação é executado pelo elemento denominado `PDPMonitor`. O `PDPMonitor` é um elemento adicionado à arquitetura tradicional de PBNM, que se encarrega de verificar o estado das condições impostas nas regras. Este elemento é fundamental tanto para o controle de execução das regras, quanto para o controle hierárquico existente entre as regras de uma mesma política, pois uma vez que uma regra de nível superior falhe (este evento deve ser percebido através do controle de execução), todas as regras imediatamente inferiores (ativas ou não) se tornam inválidas, juntamente com os monitores condicionais (`PDPMonitor`) de tais regras. O PDP e o `PDPmonitor` é desenvolvido em java. Os elementos chamados PDPs específicos (e.g. `QoSPDP`), têm a função de traduzir as políticas de alto nível, repassadas a eles através do `Internal Manager`, para que possam ser interpretadas, de acordo com cada dispositivo ou tecnologia de rede a cuja política está vinculada. O vínculo entre as políticas e as tecnologias é identificado através das variáveis que descrevem as ações e condições (e.g. a variável `qpBandwidth` está vinculada à QoS). Tais elementos também são implementados em java.

Um ambiente de testes foi montado, e reflete exatamente a o topologia apresentada Figura 6. O roteadores foram montados utilizando FreeBSD, juntamente com o `daemon routed` (que permite o roteamento através de RIP). O `daemon MRouted` foi utilizado para possibilitar a passagem de fluxos multicast, e o software `AltQ`, foi usado para fornecimento de QoS. Os testes realizados consistiram na aplicação de políticas no ambiente. O comportamento da arquitetura implementada, observado durante a aplicação dos testes, permitiu a conclusão de que é possível fornecer gerenciamento integrado de QoS e multicast, usando PBNM. Outras conclusões alcançadas serão comentadas na próxima seção.

5. Conclusões

Este trabalho descreve a pesquisa, definições e implementações feitas com o objetivo de se fornecer um mecanismo de gerenciamento capaz de integrar o gerenciamento do QoS e multicast. Após a conclusão de todas as fases, algumas conclusões foram alcançadas.

Uma das conclusões e motivações para o desenvolvimento do trabalho foi a dificuldade encontrada para fornecer um adequado gerenciamento de QoS e multicast de forma tradicional, onde são usadas diversas ferramentas, sendo que cada uma executa uma tarefa de gerenciamento e pouca ou nenhuma integração existe entre elas. Entretanto, pode-se concluir também que existem mecanismos emergentes no contexto de gerenciamento de redes, capazes facilitar a integração de qualquer tipo de gerenciamento, e neste caso específico, integração do gerenciamento de QoS e Multicast. O mecanismo de gerenciamento emergente encontrado, e já comentado anteriormente, é gerenciamento baseado em políticas (PBNM).

No entanto, durante a pesquisa, constatou-se que o gerenciamento baseado em políticas com relação à QoS já existia, e estava dando os primeiros passos no sentido de padronizar

tal gerenciamento. Portanto, o gerenciamento baseado em políticas para QoS é mais maduro. Maduro, principalmente com relação ao gerenciamento baseado em políticas para multicast, onde não foram encontrados esforços no sentido de melhor definir tal gerenciamento. Neste contexto o gerenciamento baseado em políticas para multicast necessita de maior investigação, pois esse trabalho levantou apenas alguns tópicos relativos ao gerenciamento de multicast, levando-se em conta apenas as necessidades de multicast encontradas nas aplicações avançadas, citadas no início deste artigo.

A principal e mais esperada conclusão do trabalho é que, através da arquitetura definida, aliada às decisões de opção por tecnologias a serem utilizadas na implementação do protótipo, é possível fornecer gerenciamento integrado de QoS e multicast. Tal conclusão foi obtida através da própria implementação do protótipo, assim como através dos testes realizados. Cabe lembrar que os testes realizados foram feitos sobre um ambiente de testes real, porém limitado, mas capaz de validar a implementação.

References

- Granville, L. Z. and Tarouco, L. M. R. (2001). Qame - qos-aware management environment. *25th Annual International Computer Software and Applications Conference (COMPSAC'01)*, page 269. IEEE.
- Granville, L. Z., Vaguetti, L., Almeida, M. J. B., and Tarouco, L. M. R. (2003). A pbnm system for integrated qos and multicast management. *POLICY 2003*, pages 243–246. IEEE.
- McDysan, D. (1999). *QoS and Traffic Management in IP and ATM Networks*. Number 1. McGraw-Hill Osborne Media. ISBN: 0071349596.
- Moore, B. (2003). Policy core information model (pcim) extensions. Request for Comments: 3460, Updates RFC 3060. IETF.
- Moore, B., Durham, D., Strassner, J., Westerinen, A., and Weiss, W. (2003). Information model for describing network device qos datapath mechanisms. INTERNET-DRAFT. IETF.
- Moore, B., Ellesson, E., Strassner, J., and Westerinen, A. (2001). Policy core information model – version 1 specification. Request for Comments: 3060. IETF.
- Sloman, M. (1994). Policy driven management for distributed systems. *Journal of Network and Systems Management*, 2(4):333–360. Plenum Press Publishing.
- Snir, Y., Ramberg, Y., Strassner, J., Cohen, R., and Moore, B. (2003). Policy qos information model. Request For Comments: 3644. IETF.
- Vaguetti, L., Neisse, R., Granville, L. Z., Almeida, M. J. B., and Tarouco, L. M. R. (2003). An architecture for integrated policy-based management of qos and multicast-enabled networks. *21th Simpósio Brasileiro de Redes de Computadores*, 1:217–232. Natal : UFRN/DIMAp.
- Westerinen, A., Strassner, J., Scherling, M., Quinn, B., Herzog, S., et al. (2001). Terminology for policy-based management. Request for Comments: 3198. IETF.