

Gerenciamento de Medições por Fluxo de Tráfego: Uma abordagem baseada no uso de Banco de Dados e Serviços Web

Leobino N. Sampaio¹, Lisandro Zambenedetti Granville², José A. Suruagy Monteiro¹

¹NUPERC – Universidade Salvador
R. Ponciano de Oliveira, 126 – 41950-275 - Salvador, BA

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

leobino@unifacs.br, granville@inf.ufrgs.br, suruagy@unifacs.br

Abstract. *It is a great challenge to analyze all the data obtained from the variety of measurement tools existing today. This task is even more complicated when we need to extract useful information for managing networks belonging to different domains due to systems and equipment heterogeneity. The utilization of database systems coupled with Web Services, can bring significant contribution to managing application that make use of those measurement. This article proposes a new approach for traffic flow measurement managing which provides heterogeneous systems integration using standard based technologies.*

Resumo. *Analisar os dados advindos das medições realizadas através da variada quantidade de ferramentas existentes hoje é um grande desafio. A tarefa torna-se ainda mais complicada quando se pretende extrair informações úteis ao gerenciamento das redes que pertencem a domínios diferentes. Neste sentido, a utilização de sistemas de banco de dados em conjunto com Serviços Web, pode trazer contribuições significativas às aplicações de gerenciamento que fazem uso dos dados destas medições. O presente artigo propõe uma nova abordagem para o gerenciamento das medições por fluxos de tráfego, prevendo a integração entre sistemas heterogêneos utilizando tecnologias baseadas em padrões.*

Palavras Chaves: *Gerência, Medições, Fluxos de tráfego, Banco de dados, SOA, Serviços Web.*

1. Introdução

A compreensão sobre o comportamento dos tráfegos nas redes de computadores sempre foi um desafio para pesquisadores e, principalmente, para administradores de redes. Entender o comportamento de uma rede é crítico para que o uso dos recursos possa ser melhor regido, para que os usuários possam compreender melhor o ambiente que estão utilizando, e para permitir aos administradores detectarem problemas e preverem expansões das infra-estruturas. Nesse processo, o uso de soluções para medições de rede é crítico.

Em redes locais, internas aos domínios administrativos, soluções para monitoração de redes são, hoje, amplamente empregadas, sendo que algumas chegaram até mesmo a serem padronizadas, como é o caso do RMON e RMON2 [Waldbusser, 1997], padronizadas pelo IETF (*Internet Engineering Task Force*). Nos enlaces entre domínios administrativos diferentes, entretanto, as soluções para medição são mais recentes, e por isso, menos maduras.

Nesse ambiente (enlaces entre domínios administrativos diferentes), pode-se citar, entre algumas soluções relevantes, a arquitetura RTFM [Brownlee et al., 1999] implementada pelo NetraMet [Brownlee, 1997], NetFlow [Brownlee et al., 1999] e o sFlow [Phaal et al., 2001]. Uma das características mais relevantes na arquitetura dessas soluções é a existência de um elemento coletor. Um coletor é formado, normalmente, por um host que, ligado a um equipamento de rede, coleta dados dos tráfegos para serem posteriormente analisados. Tanto o coletor, quanto o equipamento associado devem utilizar algum protocolo para troca dos dados de interesse. Para evitar uma diversidade muito grande de soluções para troca de dados entre equipamento e coletor, o IETF vem trabalhando na padronização dessa comunicação através do grupo de trabalho IPFIX (*IP Flow Information Export*) [IPFIX, 2003].

Uma questão crítica está relacionada ao método de acesso aos dados internos a um coletor. O IPFIX define a transferência de dados do equipamento para o coletor, mas não existe uma padronização para transferência de dados do coletor para outros elementos de análise. Algumas possibilidades para essa transferência são FTP (*File Transfer Protocol*), HTTP (*Hyper-Text Transfer Protocol*), etc., mas soluções diferentes podem usar protocolos diferentes. Além disso, mesmo que o protocolo fosse padronizado, existe a necessidade também de padronização da representação das informações de um coletor. Por fim, o par coletor/equipamento fornece informações relativas aos tráfegos de forma isolada: não existe uma solução capaz de controlar e correlacionar informações provenientes de vários coletores diferentes. Sendo assim, ainda que o IPFIX seja uma padronização importante, ela opera em um escopo específico, e que nem sempre é suficiente para os processos de gerenciamento da rede.

A nível mundial, as medições por fluxo também vêm sendo adotadas por iniciativas tais como a E2E piPEs (*End-to-End Performance Initiative Performance Environment System*) da Internet2 [Internet2, 2003] e INTERMON [INTERMON, 2003]. Nestes grupos já existem, inclusive, propostas de padronização dos dados resultante das medições de QoS feitas pelas diversas ferramentas. Este esforço para a padronização faz parte do escopo dos trabalhos do grupo de Serviços e Desempenho do *Global Grid Forum* [GGF, 2003], que tem como uma de suas metas implantar um ambiente integrado e colaborativo entre os variados sistemas permitindo um acesso uniforme às diversas fontes de informação. No contexto da RNP¹, através das atividades do GT-QoS [Monteiro et al., 2003], o Netflow tem sido utilizado como principal solução para identificar as características de tráfego do *backbone* nacional. Durante a sua implantação, foi possível constatar a necessidade de mecanismos para a consolidação dos dados sobre os fluxos oriundos dos diversos PoPs (pontos de presença).

Este artigo apresenta uma nova abordagem para o gerenciamento de medições por fluxo de tráfego que tem por objetivo fornecer informações a partir de um conjunto de coletores espalhados por diversos domínios administrativos. Cada coletor recebe as informações sobre os fluxos dos equipamentos gerenciados, exporta estas informações a servidores de banco de dados e que por sua vez disponibiliza através de Serviços Web (do inglês *Web Services*) [W3C, 2003b]. Além disso, o modelo apresentado é composto de três camadas (Serviços básicos, Composição e Gerenciamento). Neste documento será enfocada a primeira camada, mais especificamente o armazenamento dos dados e a sua disponibilização por meio de Serviços Web.

O restante deste artigo está assim organizado. Na seção 2 são apresentados os trabalhos relacionados e a fundamentação teórica necessária. Na seção 3 é apresentado um modelo

¹Rede Nacional de Ensino e Pesquisa - <http://www.rnp.br>

de gerenciamento e alguns de seus componentes. A seção 4 é dedicada ao protótipo e aos experimentos realizados, por fim, na seção 5 tem-se a conclusão.

2. Trabalhos Relacionados

Atualmente, manipular os dados das medições dos fluxos que passam nas redes IP é um grande desafio, dada à diversidade de ferramentas e soluções disponíveis. Mesmo com padronização da forma em que estes fluxos são medidos, ainda resta um problema referente à incompatibilidade de comunicação entre sistemas pertencentes a domínios diferentes. Estes fatores resultam nos trabalhos do grupo IPFIX e à abordagem da arquitetura SOA implementada através de Serviços Web.

2.1. IPFIX

No intuito de compatibilizar o grande número de soluções presentes em torno das medições por fluxos de tráfego, o grupo IPFIX foi criado especialmente para definir uma padronização para a coleta dos fluxos identificados pelos dispositivos de medição e exportados para os coletores. Para isso foram estabelecidas algumas atividades com o objetivo de definir, dentre outras coisas, uma arquitetura de funcionamento [Sadasivan and Brownlee, 2003], um protocolo de transporte entre medidores e coletores [Claise et al., 2003] e um modelo de dados [Calato et al., 2003].

O protocolo IPFIX, implementado sobre TCP, é responsável por transferir informações de um equipamento de rede (ex., roteador ou swicht) para um coletor. Os dados transferidos descrevem os fluxos de interesse, bem como informações complementares sobre os mesmos (ex., número de bytes, portas de transporte, endereços de origem e destino, etc.). Como abordado anteriormente, os trabalhos do IPFIX se reservam à coleta dos dados e sua transferência para os coletores. Uma vez que a coleta e exportação das informações dos fluxos estejam sendo realizadas seguindo esta padronização, outras contribuições que estão fora do escopo das atividades do grupo, ainda se fazem necessárias no que diz respeito ao armazenamento, recuperação e disponibilização dos dados.

2.2. SOA (Arquitetura Orientada a Serviços)

A Arquitetura Orientada a Serviços (do inglês *Service-Oriented Architecture* — SOA) [W3C, 2003b] [Papazoglou, 2003] trata-se de um outro tipo de abordagem no desenvolvimento de software no qual as aplicações passam a ser construídas e reorganizadas como provedoras de serviços (ou operações) específicos e bem definidos. Basicamente, fazem parte desta arquitetura três elementos: um cliente de serviços, um provedor de serviços e um agenciador de informações sobre os serviços.

- **Cliente.** O cliente é o elemento que faz a solicitação dos serviços, portanto, é quem inicia a comunicação e muitas vezes pode utilizar o atendimento dos serviços solicitados para a composição de outros serviços.
- **Provedor.** O provedor contém um ou mais serviços que são disponibilizados aos clientes. Para viabilizar o acesso aos serviços a partir dos clientes, o provedor divulga os seus serviços nos repositórios.
- **Agenciador.** Os agenciadores armazenam a descrição, classificação e localização dos serviços disponibilizados pelos provedores para todos os participantes do sistema. Os agenciadores têm um papel fundamental no funcionamento da arquitetura, pois facilita a descoberta dinâmica dos serviços por parte dos clientes.

Um dos problemas ainda críticos no SOA se dá em torno da falta de um modelo que facilite o gerenciamento, a composição e utilização dos serviços prestados pelos diversos componentes. Dentro deste escopo, Papazoglou propõe uma arquitetura estendida ao SOA (*ESOA — Extended Service-Oriented Architecture*) [Papazoglou and Georgakopoulos, 2003] na qual os elementos provedores de serviços são divididos em três camadas, que são: (1) descrição e operações básicas, (2) composição de serviços e (3) gerenciamento, como podem ser vistas na Figura 1.

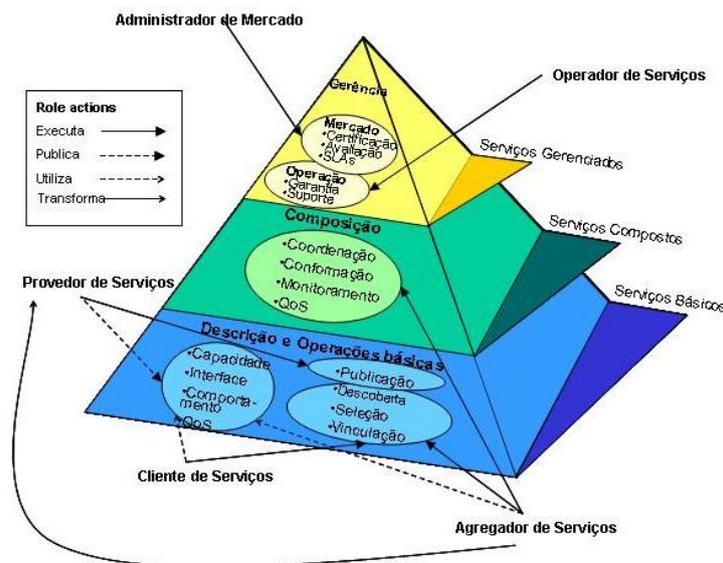


Figura 1: SOA Estendida.
[Papazoglou and Georgakopoulos, 2003]

Em resumo, a camada de descrição e operações básicas é responsável pelo fornecimento dos serviços na sua forma mais simples. Na segunda camada, os serviços oferecidos são derivados da composição dos serviços prestados pela primeira. Por fim, na última camada estão localizados os provedores de serviços mais próximos das aplicações e usuários finais. Esta arquitetura define regras e funcionalidades destinadas à consolidação de múltiplos serviços em um único, permitindo que as aplicações usuárias tenham um enfoque direcionado para a sua funcionalidade principal, descartando os detalhes de implementação de outros serviços necessários. Deste modo, essa abordagem oferece subsídios que facilitam, consideravelmente, o gerenciamento dos serviços, já que os mesmos estão classificados em três camadas.

Em termos práticos, a implementação dos conceitos e requisitos da Arquitetura Orientada a Serviços tem sido realizada através dos Serviço Web. Isto acontece simplesmente pelo fato de ser uma solução fortemente baseada na linguagem XML (*eXtended Markup Language*) [W3C, 2000], que é de ampla aceitação e tem se consolidado como a alternativa mais adequada na integração de sistemas heterogêneos.

2.3. Serviços Web

Serviços Web (WS) podem ser considerados como uma forma de *middleware* o qual possui aplicações que se comunicam por meio de protocolos tais como o HTTP. Esta característica permite que os problemas de integração, geralmente presentes no relacionamento entre as aplicações de domínios diferentes, sejam bastante reduzidos.

Tais aplicações utilizam a linguagem XML para descrever suas funcionalidades, auxiliar na invocação de seus métodos, definir tipos de dados e publicar seus serviços. Cada uma destas atividades é realizada através de tecnologias derivadas da XML, tais como a WSDL (*Web Services Description Language*) [Chinnici et al., 2003], SOAP (*Simple Object Access Protocol*) [W3C, 2003a] e o UDDI (*Universal Description, Discovery, and Integration*) [OASIS, 2002].

A WSDL é uma linguagem utilizada para descrever os serviços disponibilizados por uma aplicação. Através desta descrição, as outras aplicações podem saber a forma de interação e os protocolos necessários para a comunicação. O SOAP é o protocolo utilizado para encapsular as trocas de mensagens entre objetos WS (invocação de métodos remotos). O uso do SOAP viabiliza que aplicações possam se comunicar através da utilização de protocolos simples como o HTTP. Por fim, o UDDI é uma infra-estrutura pública de diretórios que é utilizada para publicar os serviços e a localização das aplicações.

Com base nestes aspectos, percebe-se que o WS tem uma estreita relação com a Arquitetura Orientada a Serviços. Neste caso, o agenciador de informações citado anteriormente é implementado através do UDDI, a comunicação entre os elementos ocorre via protocolo SOAP e, por fim, a forma de interação entre esses componentes é descrita em documentos WSDL. A Figura 2 apresenta uma visão resumida dos elementos envolvidos nos Serviços Web e sua relação com a SOA.

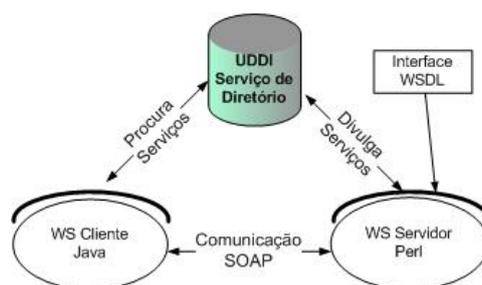


Figura 2: Arquitetura de Serviços Web.

Em termos de gerenciamento de redes, existem na literatura diversas soluções que propõem a utilização de aplicações distribuídas para a realização de medições de QoS (Qualidade de Serviço). Na maioria dos casos, por serem baseadas em tecnologias como Corba, são incapazes de trocar informações na Internet, o que não permite uma monitoração ao longo de domínios administrativos diferentes. Diante dos aspectos abordados nesta seção percebe-se que no contexto das medições por fluxo de tráfego, existe a necessidade de uma infra-estrutura que possibilite um acesso independente de plataforma e tecnologias. Neste sentido, a arquitetura SOA surge como uma boa alternativa em função da sua simplicidade e capacidade de viabilizar a interoperabilidade entre sistemas.

O gerenciamento dos dados das medições por fluxos através de WS se torna uma solução bastante conveniente, na medida em que não são exigidas grandes mudanças na infra-estrutura de rede para a sua implementação. Além de contar com a grande flexibilidade provida pela XML que fornece à aplicação a possibilidade de se adaptar a qualquer tipo de ambiente. Por estes motivos, o relacionamento dos trabalhos de padronização do IPFIX, a abordagem da arquitetura SOA estendida (ESOA) e as características do WS resulta no modelo proposto na seção seguinte, que tem como principal meta formalizar a manipulação dos dados sobre os fluxos de rede.

3. Modelo de Gerenciamento

Para as aplicações de gerenciamento que estão associadas aos usuários finais, o que interessa são as informações que podem ser disponibilizadas pelos fluxos e não os seus dados na forma em que são coletados. Atualmente, estas aplicações precisam implementar soluções próprias para ter acesso a estas informações, tendo que ter a preocupação com detalhes específicos das plataformas de cada ambiente.

Para estas aplicações, seria ideal uma solução de *middleware* que facilitasse o acesso às informações sobre os fluxos abstraindo delas os detalhes de implementação. Com este intuito, o modelo apresentado na Figura 3 sugere a criação de componentes de software que provê suas funcionalidades em forma de serviços, acessados através de interfaces padronizadas, bem definidas e de acesso uniforme. O seu funcionamento tem como base os princípios da arquitetura ESOA apresentada na seção 2.

3.1. Componentes do modelo

Levando em consideração estes aspectos, o modelo é formado por um conjunto de componentes que estão classificados em três camadas, conforme Figura 3. Em função do escopo do presente artigo, será dado um enfoque aos componentes pertencentes à primeira camada, responsáveis pelos serviços mais básicos, são eles: O coletor e o componente de banco. O primeiro é o elemento responsável por receber os fluxos exportados pelos dispositivos medidores, disponibilizar suas informações, bem como solicitar os seus armazenamentos em banco de dados. Já o segundo é o componente responsável pela manipulação das informações sobre os fluxos armazenadas em um banco de dados.

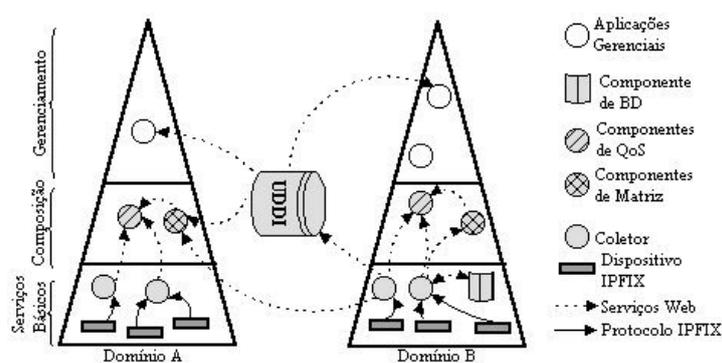


Figura 3: Modelo de Gerenciamento.

Nas bases das pirâmides da Figura 3 estão localizados os dispositivos medidores bem como os seus coletores e componentes de BD. Neste nível, as informações sobre os fluxos estão no seu formato mais primitivo. Já as aplicações pertencentes ao nível intermediário conseguem um nível de abstração um pouco maior, utilizando dados que já passaram por um processamento prévio e que são disponibilizados seguindo um padrão de acesso.

O relacionamento entre estes componentes deve acontecer de forma dinâmica, e para isso, é necessária uma infra-estrutura que permita a publicação e descoberta automática das informações sobre os serviços prestados por cada componente durante o seu funcionamento. Esta infra-estrutura é viabilizada pela adoção de um serviço de diretórios (Ex. UDDI).

3.1.1. Coletores

Neste modelo, os coletores possuem a flexibilidade de processar os dados pertencentes a um banco de dados bem como os dados que estão sendo coletados em tempo real. Este comportamento varia de acordo com os parâmetros passados na solicitação dos clientes, os quais podem especificar o intervalo de tempo dos fluxos de interesse.

O fornecimento dos dados em tempo real permite que seja feita uma análise do desempenho das aplicações no momento em que as mesmas estão utilizando a rede, viabilizando assim, a realização de ajustes sob demanda. Por outro lado, nas situações em que as informações sobre os fluxos necessitam ser armazenadas por um período de tempo maior, o uso de banco de dados se torna imprescindível. Cabe então, atribuir as funcionalidades de manipulação do banco de dados a um componente especializado e dedicado a esta finalidade. Este componente será responsável por disponibilizar serviços de acesso aos dados a qualquer componente que saiba interpretar a descrição da sua interface bem como interagir com a mesma.

3.1.2. Componentes de banco de dados

O uso de banco de dados se torna importante porque muitas vezes o volume de informações sobre os fluxos coletados pode ser muito grande, principalmente nos canais de alta velocidade. Porém, atribuir esta tarefa aos coletores pode resultar em um processamento desnecessário uma vez que nem sempre os dados coletados serão de interesse futuro.

A principal funcionalidade dos componentes de BD é facilitar o acesso aos dados sobre os fluxos, fornecendo serviços de armazenamento e recuperação de informações de interesse às aplicações usuárias. Este componente deverá conhecer a estrutura na qual os dados estão armazenados e os detalhes específicos da arquitetura do gerenciador de banco de dados, que pode variar de fabricante de acordo com o domínio em questão (ver Figura 3). Como estes componentes disponibilizam e acessam as informações de forma padronizada, a interoperabilidade entre os componentes é garantida. Esta última questão será melhor explicada na subseção seguinte.

3.2. Comunicação entre os componentes

Os componentes irão se comunicar com o propósito de prover serviços a outros componentes ou até mesmo para aplicações. Esta comunicação ocorre por meio do protocolo SOAP, e a sua forma de interação é descrita através da linguagem WSDL. Neste caso, por exemplo, quando um coletor recebe os fluxos e precisa armazenar em banco de dados, o mesmo deve possuir o conhecimento do documento WSDL do componente de banco de dados. A partir desta informação, o coletor saberá os passos necessários para ter sua demanda atendida.

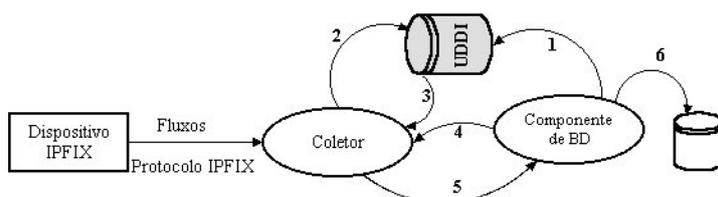


Figura 4: Comunicação entre Componentes.

Na Figura 4 é ilustrado um exemplo da comunicação entre os componentes coletores e os de banco de dados. Todo o relacionamento entre os elementos do modelo se faz através de

requisições http utilizando o protocolo SOAP para o empacotamento das mensagens, mesmo que tais elementos pertençam a domínios diferentes.

Neste exemplo, após o coletor receber as informações sobre os fluxos oriundas dos dispositivos IPFIX, o mesmo tem a possibilidade de fazer o armazenamento destas informações em banco de dados. Neste sentido, o coletor deve solicitar ao componente de BD a realização deste serviço. Para isso, é necessário que o coletor tenha o conhecimento de onde encontrar o componente de BD e como interagir com o mesmo. Sendo assim, o UDDI assume um papel fundamental. Na Figura 4, o componente de BD divulga a sua existência e os seus serviços no diretório público (PASSO 1). No PASSO 2, o coletor procura por um serviço de armazenamento em banco de dados e recebe a resposta no PASSO 3 por meio de uma URI (*Uniform Resource Identifier*). No PASSO 4, o coletor recebe o documento público WSDL, do componente de BD, descrevendo todos os detalhes necessários para a realização da comunicação. De posse deste documento e após a sua análise, o coletor está em condições de interagir com o componente de BD, e é exatamente o que acontece no PASSO 5. Por fim, após receber as mensagens dos coletores, o componente de BD armazena as informações nas suas bases de dados no PASSO 6.

Através do exemplo da Figura 4 é possível perceber algumas vantagens deste modelo de gerenciamento. Dentre elas merecem destaques: Toda a comunicação é feita por meio da linguagem XML, essa possibilidade viabiliza a implementação das aplicações mediante a conveniência dos desenvolvedores sem comprometer a interoperabilidade entre esses sistemas; Em função da extensibilidade do XML, cada um dos seus elementos poderá compor as suas funcionalidades a partir do aproveitamento de funções já disponibilizadas por outros elementos; A localização das funções dos componentes acontece de forma dinâmica, contemplando as funcionalidades do RTANS (*Real-Time Application Name Server*) apresentadas por Tham, Jiang and Ko [CK and Ko, 2000]; Como a comunicação acontece através do protocolo http, diminui a probabilidade de problemas com as políticas de segurança dos domínios; Este modelo de comunicação é bastante flexível e adaptável aos diversos ambientes de gerenciamento que tem o XML como base tecnológica.

4. Protótipo e Experimentos

4.1. Cenário do Experimento

O ambiente de testes foi iniciado com a utilização do Netflow, para identificação dos fluxos, no PoP-Ba da Rede Nacional de Ensino e Pesquisa (RNP), com a sua configuração na principal interface de entrada do tráfego do roteador (cisco série 7500).

Na configuração realizada, os fluxos foram enviados para um servidor instalado no Nuperc / UNIFACS através de um link Gigabit Ethernet, conforme apresentado na Figura 5. O servidor, da marca Gateway modelo 8400 server, possui 4 processadores Pentium III, 2Gb de memória e 7 discos SCSI de 36Gb e o sistema operacional Linux RedHat 7.3 (kernel 2.4.18-3).

4.2. Coleta dos dados

Para a coleta dos fluxos, foi utilizado o pacote flow-tools² versão 0.62. Uma vez os fluxos identificados no roteador cisco, os mesmos são transferidos via pacotes UDP para coletores. Estes dados normalmente ficariam armazenados em sistemas de arquivos porém nestes experimentos foram armazenados em banco de dados, para isso foi necessário o suporte da ferramenta flow-export do pacote flow-tools.

²<http://www.splintered.net/sw/flow-tools/>

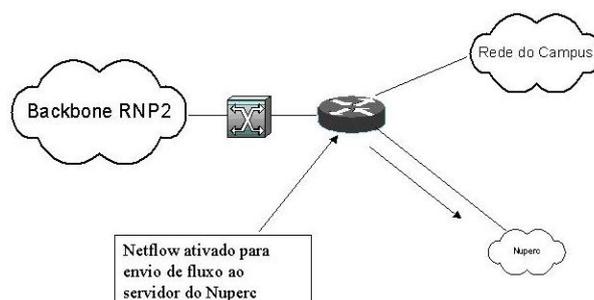


Figura 5: Cenário dos Experimentos.

4.3. Armazenamento

Para o armazenamento dos dados foi utilizado o software MySQL ³, por ser gratuito e já possuir suporte do pacote flow-tools. Mas de qualquer modo, vale ressaltar que para o modelo na seção 3, a escolha do software de gerenciamento de banco de dados não é tão relevante, já que as informações são disponibilizadas via Serviços Web. Além disso, é importante ressaltar que nestes experimentos os dados foram armazenados conforme o esquema definido em [Systems, 2001], já que para estes testes foram utilizados apenas os dados da versão 5 do Netflow.

A fim de melhorar o desempenho nas consultas aos dados sobre os fluxos foram criados índice de buscas nos campos com maior frequência de consulta, tais como IP de Origem, IP de destino e Porta.

O componente de banco de dados possui um conjunto de consultas SQL para o processamento dos dados do Netflow, estes componentes foram escritos em Java e utilizam o JDBC para acesso aos dados manipulados através do software MySQL. Estas funções, de acesso a dados, são totalmente encapsuladas, dando total transparência aos componentes que fazem uso do serviço e que por sua vez precisa apenas se preocupar com o uso das informações. Dentre as informações que são disponibilizadas por estes componentes, podem ser citadas: número total de octetos por subrede; número total de octetos por TOS (*Type of Service*); número total de octetos por porta; número total de octetos por sistema autônomo; número total em octetos por IP origem.

4.4. Disponibilização

A disponibilização dos dados armazenados em banco de dados foi realizada através do uso de Serviços Web e com base nas tecnologias de componentes Java. Para isso foi utilizado o pacote do projeto Apache Axis ⁴ o qual fornece uma boa infra-estrutura para o desenvolvimento de aplicações desta natureza. Isto por que através dele, o desenvolvedor de Serviços Web não precisa se preocupar com o tratamento de mensagens SOAP e o uso de APIs Java mais específicas como JAX-RPC (*Java API for XML-based RPC*) e JAXM (*Java API for XML Messaging*). Outra característica marcante no pacote é que uma vez criadas as classes java para a disponibilização das informações sobre os fluxos, os documentos WSDL com as descrições das interfaces são gerados automaticamente.

O Apache Axis funciona como um servlet armazenado no Tomcat. As requisições são passadas em documentos XML, empacotados em envelopes SOAP. Estas mensagens, ao chegar ao seu destino, tem seus envelopes SOAP processados pelo Axis que entrega para as

³<http://www.mysql.com>

⁴<http://ws.apache.org/axis/>

aplicações apenas o seu conteúdo. Cabe então ao desenvolvedor, apenas a tarefa de desenvolver estas aplicações.

A fim de realizar testes com a disponibilização de informações sobre os fluxos, foi desenvolvido um programa em Java que recebe como parâmetros a URI do documento WSDL do componente de banco e o nome de uma das suas funções. De acordo com estes parâmetros, o programa recupera o documento WSDL, interpreta suas informações, faz a chamada à função de interesse e por fim recupera os resultados solicitados. A Figura 6 ilustra um pequeno trecho da classe java utilizada para realizar o acesso ao componente de banco. Neste exemplo, o programa acessa a função que faz o processamento da quantidade de octetos pertencentes aos fluxos de um IP de origem num determinado intervalo de tempo (*TotalOctetsIPOrig*).

```
public class cliente1 {
    public static void main(String[] args) {
        try {
            String IPOrig = "82.82.107.225"; String t1 = "38729312"; String t2 = "49982374";
            String wsaddr = "http://200.128.80.170:8080/axis/services/ServicesBD";
            Service servico = new Service();
            Call call = (Call) servico.createCall();
            call.setOperationName(new QName(wsaddr,"TotalOctetsIPOrig"));
            call.setTargetEndpointAddress( new java.net.URL (wsaddr));
            Integer res = (Integer) call.invoke(new Object[] {new String (IPOrig), new String (t1), new String (t2)});
            System.out.println("Resultado = " + res);
        } catch (Exception e) { System.out.println(e.getMessage());}
    }
}
```

Figura 6: Trecho do código da aplicação cliente.

Na Figura 7, pode-se constatar as transferências das mensagens SOAP que foram geradas pela requisição da aplicação java e a pela resposta do componente de banco. Este monitoramento foi possível através do uso da ferramenta SOAPMonitor ⁵.

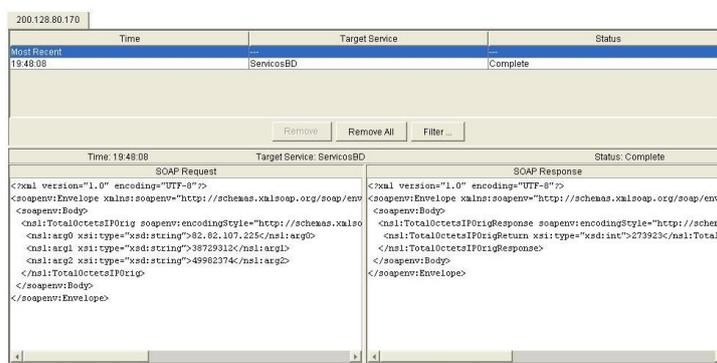


Figura 7: Mensagens monitoradas pelo SOAPMonitor.

Por fim, com o objetivo de realizar experimentos com aplicações pertencentes a domínios diferentes e verificar a interoperabilidade, foi utilizado um cliente SOAP genérico ⁶, não sendo constatado nenhum problema no acesso ao serviço disponibilizado.

4.5. Avaliação dos experimentos

Os experimentos permitiram as primeiras avaliações sobre o uso da abordagem proposta para o gerenciamento de medições por fluxo de tráfego. Foi possível perceber que o uso de Serviços Web facilita muito a integração entre as aplicações envolvidas, já que a interface padrão e o seu funcionamento pela web viabiliza a interoperabilidade entre sistemas de diferentes plataformas. Isto pode ser constatado ao fazer uso dos serviços do componente de banco de dados através de um simples programa em java.

⁵Ferramenta de monitoramento do pacote Apache Axis

⁶www.soapclient.com

Os experimentos também serviram como base de avaliação para a integração das informações sobre os fluxos às ferramentas de gerenciamento que fazem uso de Serviços Web, tais como o QAME [Granville et al., 2001]. Por fim, também foi possível perceber um sensível ganho de desempenho no acesso a essas informações com o uso de gerenciadores de banco de dados, haja visto o suporte no acesso e manipulação das informações inerentes a estes gerenciadores.

5. Conclusão

Diversos esforços internacionais vêm sendo direcionados para as medições nas redes IP. Isto fez surgir muitas propostas que, na maioria dos casos, estão dentro de um escopo específico, sem permitir a integração com outras tecnologias.

O mesmo acontece com as medições por fluxo de tráfego que tem no IPFIX o primeiro trabalho de padronização. Como resultado, esta iniciativa vai permitir que os diversos dispositivos de medição por fluxo exportem seus dados de forma padronizada sem os problemas de integração entre as diferentes ferramentas de cada fabricante. Contudo, para os fluxos medidos pelos dispositivos e enviados aos coletores, ainda se faz necessário algum modelo formal de acesso e disponibilização destes dados. Na abordagem apresentada neste artigo, o uso de Serviços Web se mostrou bastante conveniente na medida em que permite às aplicações funcionarem seguindo os princípios da Arquitetura SOA, além de ser independente de plataforma. Outra contribuição significativa trata-se do uso de sistemas gerenciadores de banco de dados para o armazenamento das informações sobre os fluxos, já que dentre outros aspectos, pode-se ter ganhos significativos de desempenho nas consultas e no tempo de desenvolvimento de aplicações avançadas.

Em suma, o armazenamento em banco de dados e a utilização de Serviços Web no gerenciamento das medições por fluxo abre novas perspectivas em torno de um gerenciamento e monitoramento de rede distribuído, rompe as barreiras implantadas pelos sistemas proprietários e faz surgir um ambiente flexível e adaptável.

Agradecemos todo o apoio que a RNP vem dando ao GT-QoS2 com o fornecimento da infra-estrutura técnica necessária para a realização dos experimentos apresentados neste trabalho.

Referências

- Brownlee, N. (1997). RFC 2123: Traffic flow measurement: Experiences with NeTraMet. Status: INFORMATIONAL.
- Brownlee, N., Mills, C., and Ruth., G. (1999). RFC 2722: Traffic flow measurement: Architecture. Status: INFORMATIONAL.
- Calato, P., Meyer, J., and Quittek, J. (2003). Information model for IP flow information export. <http://www.ietf.org/internet-drafts/draft-ietf-ipfix-protocol-01.txt>. Status: Working draft.
- Chinnici, R., Gudgin, M., Moreau, J.-J., and Weerawarana, S. (2003). Web services description language (wsdl) version 1.2 part 1: Core language. <http://www.w3.org/TR/2003/WD-wsdl12-20030611>. W3C Working draft.
- CK, T. and Ko, J. (2000). Monitoring QoS distribution in multimedia networks. *Int. J. Network Mgmt.* March/April 2000; 10: 75 – 90.

- Claise, B., Fullmer, M., Calato, P., and Penno, R. (2003). IPFIX protocol specifications. <http://www.ietf.org/internet-drafts/draft-ietf-ipfix-protocol-01.txt>. Status: Working draft.
- GGF (2003). GGF: Global Grid Forum. http://www.gridforum.org/4_GP/Perf.htm.
- Granville, L. Z., Tarouco, L. M. R., Ceccon, M. B., and Almeida, M. J. B. (2001). Integrated management of QoS-enabled networks using QAME. In Lorenz, P., editor, *Networking - ICN 2001 : First International Conference Colmar, France, July 9-13, 2001, Proceedings, Part II*, volume 2094, pages 227–290. Springer-Verlag Heidelberg.
- INTERMON (2003). INTERMON: Advanced architecture for INTER-domain quality of service monitoring, modelling and visualisation. <http://www.ist-intermon.org>.
- Internet2 (2003). E2E piPEline: End-to-end performance initiative performance environment system architecture. <http://e2epi.internet2.edu/E2EpiPEs/e2epipe11.pdf>.
- IPFIX (2003). IPFIX: IP flow information export. <http://www.ietf.org/html.charters/ipfix-charter.html>.
- Monteiro, J. A. S., Sampaio, L. N., and Figueredo, M. (2003). GT-QoS: Grupo de Trabalho de Qualidade de Serviço. <http://www.nuperc.unifacs.br/gtqos>.
- OASIS (2002). Uddi version 1 specifications. <http://www.oasis-open.org/committees/uddi-spec/doc/contribs.htm#uddiv1>.
- Papazoglou, M. (2003). Service-oriented computing: Concepts, characteristics and directions. Keynote for the 4th International Conference on Web Information Systems Engineering (WISE 2003), to appear in IEEE CS.
- Papazoglou, M. P. and Georgakopoulos, D. (2003). Introduction: Service-oriented computing. *Communications of the ACM*, 46(10):24–28.
- Phaal, P., Panchen, S., and McKee, N. (2001). RFC 3176: Inmon corporation's sflow: A method for monitoring traffic in switched and routed networks.
- Sadasivan, G. and Brownlee, N. (2003). Architecture model for IP flow information export. <http://www.ietf.org/internet-drafts/draft-ietf-ipfix-arch-02.txt>. Status: Working draft.
- Systems, C. (2001). NetFlow Services Solutions Guide. <http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.htm>. White Paper.
- W3C (2000). Extensible markup language (XML) 1.0 (second edition). <http://www.w3.org/TR/2000/REC-xml-20001006.pdf>. W3C Recommendation.
- W3C (2003a). Soap version 1.2 part 0: Primer. <http://www.w3.org/TR/2003/REC-soap12-part0-20030624>. W3C Recommendation.
- W3C (2003b). Web services architecture. <http://www.w3.org/TR/ws-arch/>. Status: Working draft.
- Waldbusser, S. (1997). RFC 2021: Remote network monitoring management information base version 2 using SMIV2. Status: PROPOSED STANDARD.