

Comparação e Avaliação dos Protocolos NETCONF e SOAP para Configuração de Dispositivos

Rodrigo S. Alves, Fábio V. Hecht, Rafael Telöken, Lisandro Z. Granville,
Maria Janilce B. Almeida, Liane Margarida R. Tarouco

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

{sanger, fhecht, rteloken, granville, janilce, liane}@inf.ufrgs.br

***Abstract.** Device configuration is an important management task that needs proper protocols that accomplished its basic requirements. Considering the current available configuration protocols, NETCONF is distinguished from others. On the other hand, SOAP can also be used in configuration and today pl/ays an important role on Web Services popularization, technology that is responsible for Web applications interoperability. This paper discusses the usage of NETCONF and SOAP for device configuration. In addition, we present system prototypes that implement such protocols and allow their performance evaluation concerning response time and bandwidth consumption.*

***Resumo.** A configuração de dispositivos é uma importante tarefa de gerenciamento que requer protocolos que atendam suas necessidades básicas. Dentre os protocolos de configuração disponíveis na atualidade, destaca-se o NETCONF. Por outro lado, SOAP também pode ser usado para configuração e vem ganhando importância com a atual popularização dos Web Services, os quais proporcionam interoperabilidade entre aplicações Web. Neste artigo é discutido o uso de NETCONF e SOAP para a configuração de dispositivos. Além disso, são apresentados protótipos que implementam tais protocolos e que permitiram a realização de avaliações de desempenho dos mesmos em relação ao tempo de resposta e banda consumida.*

1. Introdução

A configuração de dispositivos é uma tarefa crítica de gerenciamento, pois envolve alterações no estado da rede, da qual, cada vez mais, se exige um funcionamento com garantias de qualidade de serviço (QoS) e com um menor número possível de falhas ou interrupções. Por esse motivo, evidencia-se a importância do uso de protocolos adequados à tarefa de configuração. Dentre os requisitos de um protocolo para esse fim, pode-se citar, entre outros, suporte a transações, controle de conflitos, notificação de erros e capacidade de *rollback* [MacFaden et al. 2003].

Atualmente, uma opção natural de protocolo de configuração é o SNMP (*Simple Network Management Protocol*), largamente difundido entre fornecedores e, portanto, frequentemente encontrado em dispositivos comerciais. Porém, para ser utilizado concretamente como protocolo de configuração, o usuário do SNMP precisa levar em consideração as carências do protocolo que devem ser resolvidas nos níveis do projeto

da MIB (*Management Information Base*), do agente SNMP e da própria aplicação gerente. Tais carências do SNMP são apresentadas (e soluções são propostas) pelo grupo de trabalho SNMPConf (*Configuration Management with SNMP*) do IETF (*Internet Engineering Task Force*) [MacFaden et al. 2003].

Dentro do próprio IETF, entretanto, existe um outro grupo de trabalho, chamado NetConf (*Network Configuration*), que propõe o uso de um novo protocolo de configuração, também chamado NETCONF [Enns 2004], cujo objetivo é o de unificar a maneira pela qual os dispositivos de rede são configurados, propondo padrões que sejam simples e genéricos o suficiente de forma a abranger todos os tipos de dispositivos. De forma geral, pode-se dizer que o NETCONF é um conjunto de definições de documentos XML (*Extensible Markup Language*) a serem utilizadas na interação gerente-agente.

Em paralelo com a necessidade de um protocolo de configuração, a tecnologia de Web Services [Curbera et al. 2002] tem sido alvo de grande investigação no contexto de gerenciamento de redes. Dentre as características dos Web Services pode-se citar sua simplicidade e padronização, que contribuem para a obtenção da interoperabilidade entre aplicações. O protocolo largamente utilizado para troca de mensagens Web Services é o SOAP (*Simple Object Access Protocol*) [Mitra 2003], que, assim como o NETCONF, também é baseado em XML.

NETCONF e SOAP possuem uma intersecção interessante: o IETF propõe que uma das formas de transportar mensagens NETCONF seja encapsulando tal protocolo em mensagens SOAP [Goddard 2004]. Outros encapsulamentos também são propostos como, por exemplo, NETCONF sobre BEEP [Lear e Crozier 2004] e NETCONF sobre SSH [Wasserman e Goddard 2004]. É curioso, por outro lado, perceber que o próprio SOAP pode ser utilizado como protocolo para configuração de redes e, portanto, não seria necessária a existência do NETCONF.

O objetivo deste artigo é avaliar o uso de NETCONF e SOAP no contexto de configuração de redes. Para tal, são considerados 4 cenários de gerenciamento utilizando arquiteturas de protocolos distintas. Em particular, estamos interessados em verificar o desempenho do NETCONF e SOAP em relação ao consumo de banda e ao tempo de resposta dos protocolos ao consultar a tabela de roteamento de um dispositivo de teste baseado em Linux. É importante notar que, pelo fato da proposta do protocolo NETCONF ser recente, tem-se poucos (ou talvez nenhum) resultados a respeito do desempenho do NETCONF e seus possíveis encapsulamentos.

O restante deste trabalho está organizado como segue. A seção 2 apresenta o NETCONF e seu encapsulamento sobre SOAP. A seguir, na seção 3, são descritas implementações que utilizam tanto NETCONF como SOAP no gerenciamento de rotas de um dispositivo Linux de teste. Na seção 4 são realizados experimentos e os resultados obtidos com a comparação de desempenho dos protocolos são mostrados. Por fim, são apresentados os trabalhos futuros e as conclusões deste artigo na seção 5.

2. Gerenciamento de Redes via NETCONF e SOAP

Nesta seção o protocolo NETCONF é rapidamente apresentado, o uso de Web Services no contexto de gerenciamento de redes é revisto e, principalmente, o encapsulamento de mensagens NETCONF sobre SOAP é descrito.

2.1. NETCONF

O NETCONF é uma proposta de protocolo de configuração que ainda está em fase de padronização dentro do IETF. Porém, suas principais funcionalidades já se encontram documentadas no *draft* que o define, documento este que já está em sua quarta versão. A pretensão é de que esse protocolo unifique a maneira com que dispositivos são configurados. Sua proposta é definir um conjunto de instruções uniforme para serem disponibilizadas por vários tipos de dispositivos e de fabricantes. Essas instruções definem mecanismos para obter, instalar, manipular e remover configurações e estatísticas de dispositivos.

O princípio de funcionamento do NETCONF é baseado no paradigma RPC (*Remote Procedure Call*), através do qual é definido um conjunto de operações do protocolo. Resumidamente, um cliente (gerente NETCONF) codifica uma requisição RPC em XML e a envia ao servidor (agente NETCONF). O servidor, ao receber uma mensagem NETCONF, processa a requisição e envia uma resposta RPC de volta ao cliente também codificada em XML. Tanto a requisição quanto a resposta em XML têm suas estruturas totalmente descritas em XML *schema*, permitindo a ambas as partes (gerente a agente NETCONF) validarem as mensagens trocadas.

O NETCONF define um pequeno conjunto de operações básicas para obter dados de estado e gerenciar configurações de dispositivos de rede. Além das operações básicas, operações adicionais podem ser definidas, dependendo das capacidades anunciadas pelo dispositivo gerenciado. Dentre as operações básicas do protocolo, pode-se citar as abaixo:

- <get-config> para obter uma configuração;
- <edit-config> para alterar uma configuração;
- <copy-config> para copiar uma configuração;
- <delete-config> para remover uma configuração;
- <lock> para requisitar exclusividade no acesso a algum dado;
- <unlock> para liberar a exclusividade de acesso sobre algum dado.

A listagem abaixo apresenta uma requisição e uma resposta NETCONF para realizar a cópia da configuração em execução no sistema <running> para um arquivo texto qualquer (*teste.txt*). A origem dos dados a serem copiados é indicada pelo elemento <source> e o destino pelo elemento <target>.

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <copy-config>
    <source>
      <running/>
    </source>
    <target>
      <url>ftp://example.com/configs/teste.txt</url>
    </target>
  </copy-config>
</rpc>
```

```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

O NETCONF não define nem sugere o uso de um protocolo de aplicação específico para transportar suas mensagens, porém, impõe requisitos básicos que tais protocolos devem preencher, como, por exemplo, possuir suporte a conexões e possibilitar autenticação e criptografia dos mesmos.

A definição original do NETCONF considera sua implementação diretamente sobre TCP, porém, como citado anteriormente, existem *drafts* do IETF que definem como o protocolo deve ser encapsulado sobre BEEP, SSH e SOAP. Nas avaliações do NETCONF apresentadas neste artigo serão considerados os encapsulamentos do NETCONF sobre TCP e sobre SOAP, o que será apresentado a diante.

2.2. Web Services para Gerenciamento de Redes

Unindo RPC e XML, o protocolo SOAP (*Simple Object Access Protocol*) surgiu como uma opção para troca de mensagens entre aplicações na Web. O protocolo pode ser transportado sobre uma variedade de protocolos, mas sua implementação sobre HTTP é a mais largamente utilizada. Atualmente, SOAP ganhou destaque por ser o protocolo central da arquitetura de Web Services, cada vez mais utilizada para a interoperação entre aplicações.

O uso de Web Services no contexto de gerenciamento de redes ganhou especial atenção da comunidade científica nos últimos três anos, com investigações relacionadas ao consumo de banda e de tempo de resposta [Neisse et al. 2004]. No contexto da indústria, dois consórcios principais se destacam por trabalhar na definição de padrões de Web Services para gerenciamento de redes. O primeiro consórcio, chamado de OASIS [Oasis Consortium 2004], procura não apenas verificar como Web Services podem ser utilizados para gerenciamento de redes, mas também como os Web Services devem ser gerenciados. Recentemente, um segundo consórcio formado por Microsoft, Intel, Dell e AMD lançou uma especificação chamada WS-Management [Arora 2004] que define como mensagens SOAP devem ser codificadas ao se fazer gerenciamento de redes através deste protocolo.

É importante notar que tanto o uso de Web Services para gerenciamento de redes, quanto à aplicação de NETCONF para configuração de dispositivos são propostas alternativas ao SNMP, que possui, como citado, restrições críticas para configuração. Apesar disso, NETCONF e SOAP seguem também o modelo cliente-servidor, ou gerente-agente, também adotado no SNMP. Porém as diferenças mais importantes estão no fato de que NETCONF e SOAP são baseados em documentos XML que são textuais, enquanto SNMP é baseado em mensagens binárias bem mais curtas. Além disso, SNMP utilizada como protocolo de transporte o UDP, enquanto NETCONF e SOAP podem ser encapsulados, normalmente, em protocolos de aplicação como HTTP e HTTPS. O mais interessante, por fim, é o fato de que NETCONF e SOAP podem co-existir quando mensagens NETCONF são encapsuladas sobre SOAP.

2.3. NETCONF sobre SOAP

Existe atualmente no IETF um *draft* denominado “*Using the Network Configuration Protocol (NETCONF) Over the Simple Object Access Protocol (SOAP)*” que define a implementação do NETCONF sobre SOAP. Nesse documento são apresentadas recomendações e exemplos de NETCONF sobre SOAP e é argumentado que SOAP é adequado para esse fim principalmente por também operar conforme o paradigma RPC.

O uso de XML, a característica RPC e a popularidade do uso de Web Services tornam natural considerar SOAP como protocolo de aplicação para transportar as mensagens do NETCONF. Além disso, a implementação sobre SOAP traz muitos benefícios, como grande quantidade de ferramentas de desenvolvimento disponíveis e integração com sistemas já utilizados. Entretanto, como as mensagens SOAP são codificadas em XML, que é textual, elas são maiores do que suas equivalentes em formatos binários como CORBA [Raj 2004] e DCOM [DCOM 2004]. A listagem abaixo apresenta um exemplo de encapsulamento NETCONF sobre SOAP, que corresponde a solicitação e resposta de configuração anteriormente mostrada.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <rpc id="101"
      xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <copy-config>
        <source>
          <running/>
        </source>
        <target>
          <url>ftp://example.com/configs/teste.txt</url>
        </target>
      </copy-config>
    </rpc>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <rpc-reply id="101"
      xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
    </rpc-reply>
  </soapenv:Body>
</soapenv:Envelope>
```

Embora SOAP possa ser implementado sobre vários protocolos (ex.: BEEP, SMTP), na prática as implementações atuais utilizam HTTP ou HTTPS. Ainda que existam críticas ao uso de SOAP e HTTP para o transporte de mensagens NETCONF, tal combinação é factível e mostra-se interessante por proporcionar facilidades de implementação.

3. NETCONF e Web Services para Configuração de Dispositivos

Tanto NETCONF quanto SOAP definem duas entidades principais: gerentes e agentes (como no SNMP). O gerente é o software que solicita a execução de uma ação (ex.: de leitura ou escrita) a um agente localizado em uma máquina remota. O agente recebe as solicitações do gerente, efetua as ações necessárias e responde informando o resultado da operação requisitada. Nesta seção são descritos quatro ambientes práticos utilizados para a configuração de dispositivos via NETCONF e SOAP.

3.1. Agente NETCONF sobre TCP

NETCONF sobre TCP é um encapsulamento natural onde o protocolo de aplicação (NETCONF) utilizada o protocolo de transporte (TCP) para trocar mensagens entre um cliente e um servidor, ou, no contexto de gerenciamento, entre um gerente e um agente.

O software Yenca [YENCA 2004] foi a primeira implementação de NETCONF sobre TCP disponibilizada à comunidade de gerenciamento. Lançado em março de 2004, esse software foi desenvolvido no laboratório LORIA, na França e inclui no seu pacote de distribuição um agente NETCONF, desenvolvido em C, e um gerente NETCONF, desenvolvido em Java. A Figura 1 abaixo apresenta a arquitetura de protocolos utilizada quando do encapsulamento do NETCONF sobre TCP.

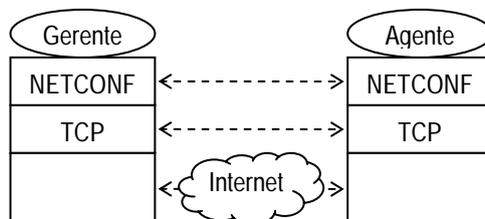


Figura 1. Encapsulamento NETCONF sobre TCP

Na versão atual, o agente Yenca possui suporte apenas para o gerenciamento de interfaces e tabelas de roteamento. Os autores do software consideram que o suporte a interfaces e leitura da tabela de roteamento está consistente, mas que a configuração das tabelas precisa ser melhorada. O agente Yenca pode ainda ser estendido através de módulos extras de gerenciamento. Tais módulos são carregados pelo agente em tempo de execução, não sendo necessário recompilar o mesmo para adicionar novos módulos.

Durante os testes realizados (apresentados nas próximas seções), foram encontradas incompatibilidades da implementação do Yenca com o *draft* do NETCONF. Entre elas, pode-se citar o fato de que os exemplos mencionados no documento do IETF incluem um atributo `xmlns` que define o *namespace* utilizado. Porém, o agente Yenca falha quando esse atributo é usado nas mensagens NETCONF.

3.2. Gateway NETCONF sobre SOAP

Como citado anteriormente, um dos *drafts* do IETF apresenta o encapsulamento de NETCONF sobre SOAP, e motiva tal encapsulamento não apenas com vantagens técnicas, mas, principalmente, com vantagens práticas. A principal dessas vantagens é o fato de um protocolo de gerenciamento, ao utilizar SOAP, ter maior probabilidade de ser implementado e adotado, dada sua larga aceitação por parte de importantes empresas e fabricantes, e dado seu uso atual em uma grande variedade de sistemas. Além disso, é suposto no *draft* que o encapsulamento do próprio SOAP é realizado sobre HTTP ou HTTPS, visto que este é seu encapsulamento mais freqüente.

Atualmente, de acordo com o conhecimento dos autores, não existe nenhuma implementação de NETCONF sobre SOAP, provavelmente devido ao fato dos *drafts* do IETF serem extremamente recentes. Considerando que o software Yenca é uma implementação disponível, mas que força o encapsulamento de NETCONF diretamente sobre TCP, um *gateway* NETCONF/SOAP para NETCONF/TCP foi desenvolvido.

O *gateway* NETCONF/SOAP para NETCONF/TCP tem a função de receber solicitações NETCONF encapsuladas sobre SOAP e encaminhá-las ao agente NETCONF do Yenca através de encapsulamento direto sobre TCP. Esse *gateway*, portanto, permite a comunicação entre um gerente operando via NETCONF sobre SOAP com um agente Yenca (NETCONF sobre TCP). A Figura 2 apresenta a arquitetura de protocolos considerando este segundo cenário.

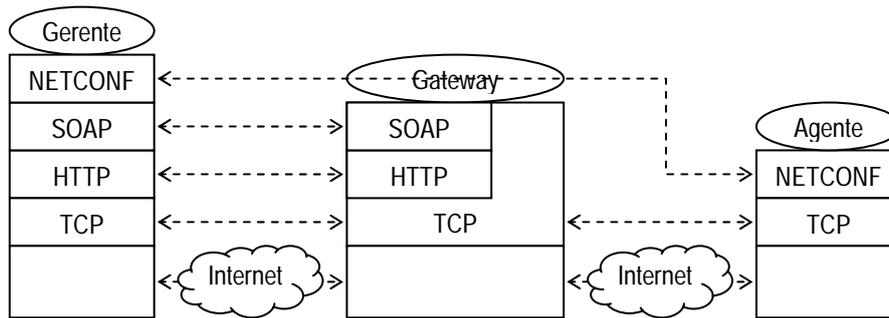


Figura 2. Arquitetura de protocolos com o gateways

Como pode ser visto, o *gateway* opera interagindo com SOAP (na comunicação com o gerente), e interagindo com o TCP (na comunicação com o agente). O *gateway* foi implementado em PHP [PHP 2004] utilizando a biblioteca nuSOAP [Ayala 2004] e MiniXML [MiniXML 2004]. A escolha de PHP deve-se ao fato deste ser uma linguagem flexível e gratuita. O nuSOAP é uma API para Web Services em PHP bastante utilizada pela comunidade de software livre, e foi escolhida por ser relativamente rápida se comparada com outras APIs disponíveis [Pear 2004].

O *gateway* opera retirando do envelope SOAP a mensagem NETCONF e repassando a mesma para o Yenca via TCP. Quando o Yenca responde a requisição com uma mensagem NETCONF, então o *gateway* monta o envelope SOAP e responde ao gerente. Na implementação corrente, devido a falta de conformidade total do Yenca com o *draft* do NETCONF, foi preciso, dentro do *gateway* processar as mensagens NETCONF antes de as mesmas serem repassadas. Especificamente, como o Yenca não aceita o atributo `xmlns` do elemento `<rpc>`, mas esse é explicitamente definido no *draft* que apresenta o encapsulamento do NETCONF sobre SOAP, foi preciso realizar um *parsing* do XML proveniente do gerente para remover esse atributo.

3.3. Agente NETCONF sobre SOAP

A implementação de NETCONF sobre SOAP proporcionada pelo *gateway* apresentado na seção anterior requer a instalação de dois softwares: o Yenca e o próprio *gateway*. Além disso, um novo elemento de software (o *gateway*) entre agente e gerente introduz, inevitavelmente, um *overhead* de processamento. Assim, num terceiro cenário o agente Yenca foi completamente substituído por um agente NETCONF sobre SOAP.

Para a implementação do agente, novamente, foi utilizada a linguagem PHP, a biblioteca nuSOAP e a biblioteca MiniXML, que foi utilizada para gerar e analisar gramaticalmente mensagens XML do NETCONF. A Figura 3 apresenta a arquitetura de protocolos considerando o novo agente NETCONF criado.

Tanto no *gateway* apresentado anteriormente, quanto no novo agente NETCONF, o código-fonte da biblioteca nuSOAP teve de ser adaptado para que as mensagens SOAP geradas seguissem fielmente os padrões de mensagem indicados pelo IETF. As alterações estão principalmente relacionadas com os cabeçalhos HTTP. Por exemplo, diretivas de controle de *caches* HTTP tiveram de ser manipuladas de forma que agentes e gerentes (e *gateway*, no cenário anterior) NETCONF não armazenassem mensagens em suas eventuais *caches* locais. Por isso, cabeçalhos HTTP proibindo *cache* foram inseridos. Além disso, foi desligado o uso de *stuffing* de caracteres por

parte do nuSOAP, uma vez que mensagens NETCONF também são documentos XML consistentes que devem ser enviados sem alteração de caracteres típicos da linguagem como os sinais de menor (<) e maior (>).

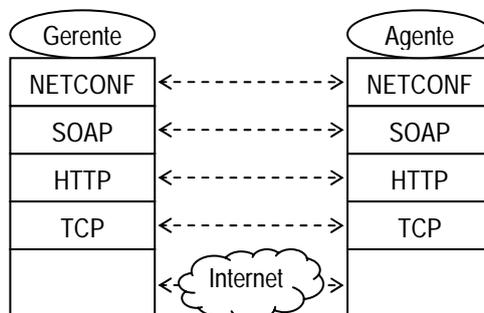


Figura 3. Arquitetura de protocolos para o novo agente NETCONF sobre SOAP

O suporte de gerenciamento do novo agente se restringe, na versão atual, ao gerenciamento de interfaces e tabela de roteamento, assim como o Yenca, visto que para os propósitos de avaliação a serem apresentados a seguir, interfaces e tabela de roteamento são suficientes. Entretanto, é importante ressaltar que enquanto no Yenca as informações de gerenciamento são coletadas através de software desenvolvido na linguagem C, no novo agente NETCONF sobre SOAP as mesmas informações são coletas no sistema final através de *scripts* PHP, e que, por se tratar de uma linguagem interpretada, apresenta um desempenho inferior em relação à linguagem C.

3.4. Agente SOAP

Como citado, SOAP pode ser utilizado como um protocolo para configuração de redes. Assim, o terceiro cenário é composto por um novo agente de gerenciamento implementado diretamente sobre SOAP, e que expõe exatamente as mesmas operações NETCONF, apesar de não codificar as solicitações e respostas através deste protocolo.

Nessa última implementação que também utiliza nuSOAP foi criada uma operação Web Service que realiza a recuperação da tabela de roteamento. Um cliente (gerente) invoca esta operação remotamente em um Web Service (agente), o qual obtém as informações junto ao dispositivo, e devolve a resposta utilizando SOAP. A Figura 4 apresenta a arquitetura de protocolos deste último cenário.

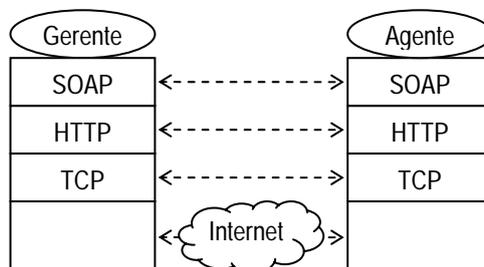


Figura 4. Arquitetura de protocolos para o gerente e agente SOAP

É importante notar que neste último cenário nem gerente nem agente necessitam manipular documentos XML, como acontecia nos cenários anteriores. Isso é decorrência do fato de a API nuSOAP já fazer esta manipulação de mensagens e entregar, para os processos gerente e agente, dados já manipulados.

3.5. Gerente

Para efetuar a interação com os agentes (e *gateway*) apresentados anteriormente e permitir a realização dos testes de desempenho desejados, um gerente de configurações foi implementado. Tal gerente é uma aplicação baseada na Web implementada também em PHP e nuSOAP. Para o suporte ao NETCONF, módulos específicos foram desenvolvidos. Apesar de o pacote de distribuição do Yenca já fornecer um gerente Java, optou-se por utilizar o novo gerente desenvolvido para homogeneizar o acesso aos 4 cenários anteriores a partir de um mesmo sistema origem. Além disso, como o gerente Yenca só tem suporte ao NETCONF sobre TCP, este não poderia ser utilizado.

Um processo simples de autenticação do operador permite restringir o acesso ao gerente. Após a autenticação, o usuário escolhe o tipo de operação de gerenciamento que deseja realizar, bem como indica qual o encapsulamento de protocolos a ser utilizado. Ao contactar o agente remoto, o usuário deve informar também o nome ou endereço IP da máquina que hospeda tal agente. A Figura 5 apresenta a interface de consulta via NETCONF ou SOAP.

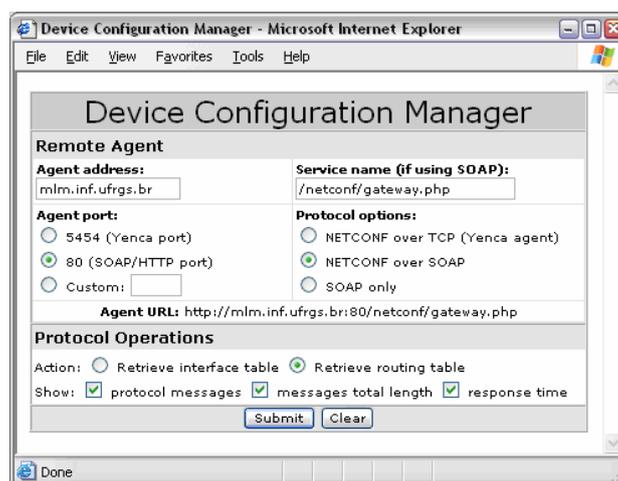


Figura 5. Interface de solicitação da aplicação gerente

Após o usuário requisitar uma operação, o gerente busca as configurações da tabela de roteamento e/ou a lista de adaptadores de rede. Para tanto, envia uma requisição usando o encapsulamento de protocolos adequado. A Figura 6 apresenta as mensagens trocadas em uma consulta e ao gateways NETCONF sobre SOAP para a obtenção das interfaces de rede da máquina mlm.inf.ufrgs.br.

4. Avaliação

Para a avaliação dos encapsulamentos de protocolos apresentados anteriormente foi montada uma rede de teste composta por duas máquinas interligadas diretamente via cabo cruzado. Uma das máquinas hospedou o gerente e a outra o agente e o *gateway*. As máquinas possuem processador AMD K6 233 MHz, com 64 MB de memória RAM, sistema operacional Linux e distribuição Fedora Core 2. O enlace entre as máquinas é de 100 Mbts. Os *softwares* utilizados foram o Apache 2.0, o PHP 4.3.4 e o Yenca.

Os testes foram realizados utilizando quatro configurações distintas do dispositivo Linux gerenciado:

- Somente agente Yenca;
- Agente Yenca com *gateway* NETCONF sobre SOAP;
- Somente agente NETCONF sobre SOAP;
- Somente agente SOAP.



Figura 6. Mensagens de solicitação e resposta no gerente

Três parâmetros foram analisados: o tempo de resposta, a quantidade de bytes transferidos e a banda consumida. Os testes consideraram a recuperação de tabelas de roteamento com 4, 5, 6, 7, 10, 15, 20, 30 e 40 rotas.

4.1. Tempo de Resposta

Para cada tamanho de tabela de roteamento e para cada configuração do dispositivo gerenciado, foram realizadas 30 medições, com as quais foi calculada a média e o desvio padrão para o tempo de resposta. O intervalo de confiança das amostras é de 95%. A coleta dos dados ocorreu no gerente, disparando um contador de tempo logo antes da requisição e parando-o mediante a chegada da resposta. O gráfico da Figura 7 apresenta os resultados de tempo de resposta obtidos com as implementações apresentadas anteriormente.

O agente Yenca mostrou-se mais eficiente, com tempo de resposta médio abaixo de 0,1 segundo para todos os números de rotas testados. O *gateway* NETCONF/SOAP apresentou um desempenho inferior (cerca de 12,5 vezes mais lento que o Yenca), resultado esse devido, principalmente, ao *overhead* de intermediação de mensagens.

O agente NETCONF/SOAP, escrito em PHP, mostrou-se significativamente mais lento, especialmente quando o número de rotas é maior, demonstrando pouca escalabilidade. Esse problema é devido principalmente à realização de *parsing* XML no nível do PHP, que é uma linguagem interpretada (tal tarefa é realizada no Yenca utilizando a linguagem C). No agente SOAP não há a necessidade de realizar *parsing* da mensagem NETCONF, visto que o protocolo não é utilizado e suas mensagens não precisam ser entendidas e montadas. A necessidade de realização de *parsing* e a forma como ele é implementado são determinantes para o tempo de resposta obtido nos testes.

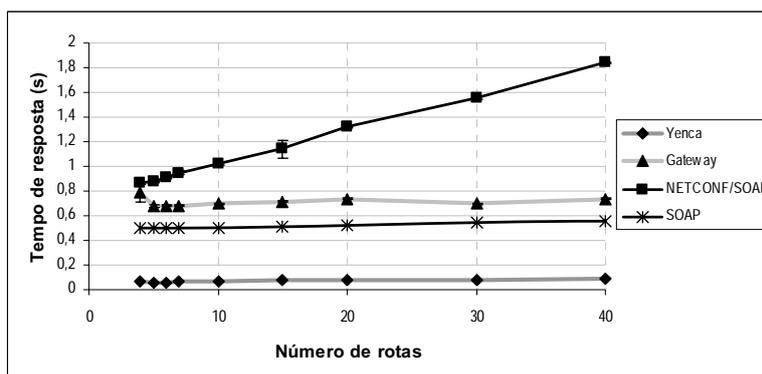


Figura 7. Gráfico comparativo do tempo de resposta

A parcela de tempo despendida com a atividade do *gateway* pôde ser avaliada pela diferença entre os tempos de resposta do Yenca e do *gateway* NETCONF/SOAP. Tal parcela foi, em média, cerca de 0,65 segundo. Além disso, a diferença entre os tempos obtidos é devida também ao *overhead* de transmissão dos dados pelo nuSOAP.

Com exceção do agente NETCONF sobre SOAP, os outros agentes mantiveram o tempo de resposta relativamente estável, apresentando um acréscimo baixo em função do aumento no número de rotas na tabela de roteamento.

4.2. Tráfego gerado

O tráfego gerado pelas implementações foi observado através da monitoração do enlace entre gerente e agente utilizando o *sniffer* Ethereal [Ethereal 2004]. O tráfego total considera não apenas as mensagens NETCONF e SOAP, mas também todos os bytes gerados pelos níveis inferiores das hierarquias de protocolos (ex.: HTTP, TCP, IP e ethernet). A Figura 8 apresenta os resultados dessas medições.

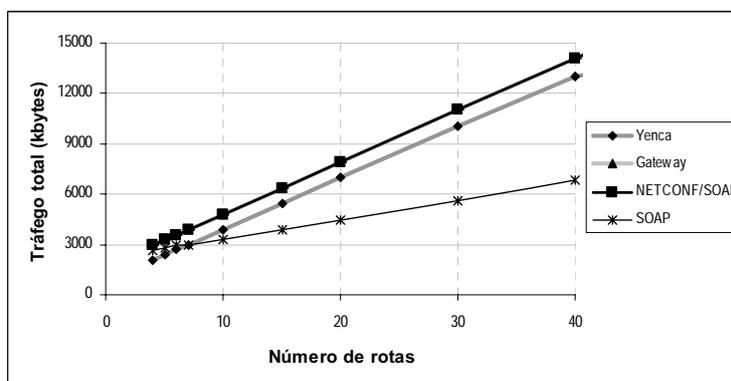


Figura 8. Total de tráfego gerado

Analisando os resultados percebe-se que as curvas correspondentes ao tráfego gerado pelo *gateway* e pelo agente NETCONF/SOAP estão sobrepostas. Isso se deve ao fato de ambos utilizarem NETCONF sobre SOAP como protocolo de aplicação.

Como era esperado, o agente SOAP gerou menos tráfego do que as implementações baseadas em NETCONF sobre SOAP (*gateway* e agente NETCONF/SOAP) justamente por não usar NETCONF. Porém, um resultado importante a ser considerado é que o agente SOAP gerou menos tráfego que o agente Yenca. O Yenca utiliza NETCONF sobre TCP diretamente, enquanto o agente SOAP utiliza SOAP sobre HTTP sobre TCP. Isso evidencia o considerável *overhead* inserido pelo NETCONF, uma vez que a implementação que utiliza HTTP e SOAP, presumidamente, deveria gerar mais tráfego por possuir um número maior de encapsulamentos. Por fim, o agente Yenca, por não utilizar SOAP, gerou menos tráfego que as implementações do *gateway* e do agente NETCONF/SOAP, o que era esperado.

Com o aumento do número de rotas, o acréscimo de tráfego gerado permaneceu constante. Pôde-se observar que NETCONF sobre SOAP gera, em média, 19% mais tráfego que NETCONF diretamente sobre TCP.

4.3 Consumo de banda

A banda consumida foi obtida somando os bytes gerados em uma requisição-resposta e dividindo esse valor pelo tempo de resposta. A Figura 9 apresenta esses resultados.

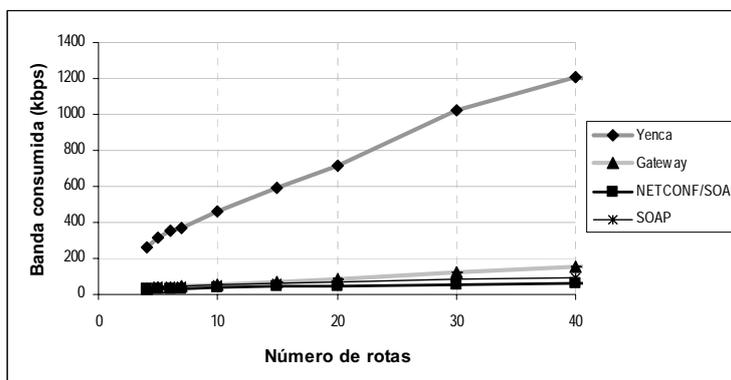


Figura 9. Banda consumida

Os testes realizados com o agente Yenca apresentaram os maiores índices de banda consumida. Isso se deve ao fato do Yenca apresentar um tempo de resposta consideravelmente menor do que os outros, e, em conseqüência, colocar mais informações na rede em menos tempo. Portanto, o menor tempo de processamento do agente Yenca resultou em um valor alto de banda consumida.

O *gateway* PHP apresentou o menor consumo de banda, de 47,5 kbps em média. Já o agente NETCONF/SOAP foi o que menos banda consumiu, o que se justifica pelo alto tempo de resposta. Em média, foram consumidos 22 kbps de banda. Por fim, verificou-se que o agente SOAP consumiu menos banda que o *gateway*, além de ter obtido menor tempo de resposta e ter gerado menos tráfego.

5. Conclusões e Trabalhos Futuros

Este artigo apresentou a avaliação de aspectos de desempenho na utilização dos protocolos NETCONF e SOAP (e seus encapsulamentos) para a configuração de redes.

A definição do NETCONF ainda não está concluída pelo IETF, porém esse protocolo se mostra como uma alternativa interessante para o gerenciamento de configurações. Por outro lado, o uso de Web Services (via SOAP) também é uma alternativa que ainda não havia sido comparada.

Enquanto o NETCONF é um protocolo específico para configurações, o SOAP é um protocolo genérico para realizar chamadas remotas de procedimentos (RPC). Ambos podem ser encapsulados em protocolos diferentes, formando arquiteturas de redes distintas. Em geral, entretanto, SOAP é encapsulado em HTTP, e NETCONF, de acordo com o IETF, pode ser encapsulado direto sobre TCP, BEEP, SSH e, inclusive, SOAP.

Uma questão importante que normalmente surge neste contexto é a da real necessidade de um novo protocolo de configuração (NETCONF) mediante a existência de um protocolo de uso geral já amplamente aceito (SOAP). As avaliações apresentadas neste artigo não ajudam a justificar a existência do NETCONF.

Quatro arquiteturas de protocolos foram avaliadas: NETCONF sobre TCP (via agente Yenca), NETCONF sobre SOAP via (*gateway* e via um novo agente), e apenas SOAP (através de um agente SOAP). Os encapsulamentos mais “pesados” (NETCONF sobre SOAP), como esperado, apresentaram o pior desempenho, tanto em relação ao tempo de resposta quanto ao consumo de banda. Entretanto, o resultado mais interessante está relacionado com o agente SOAP e a versão mais “leve” do NETCONF (NETCONF sobre TCP). Supostamente, o desempenho de SOAP deveria ser pior que o do NETCONF sobre TCP, uma vez que mensagens SOAP ainda são encapsuladas em HTTP e, finalmente, sobre TCP. O tempo de resposta no caso do agente SOAP foi relativamente mais alto que o Yenca (NETCONF sobre TCP), mas isso se justifica porque o agente SOAP foi implementado em PHP (uma linguagem interpretada), enquanto que o Yenca é implementado usando C (uma linguagem compilada).

Por outro lado, o tráfego gerado pelo agente SOAP foi menor que o tráfego gerado pelo agente Yenca, ainda que nesse último caso existisse um menor número de protocolos encapsulados. Com isso, é possível afirmar que o *overhead* do protocolo NETCONF é, sozinho, maior que o *overhead* de SOAP sobre HTTP. Como funcionalmente tanto NETCONF quanto SOAP são capazes de fornecer operações de configuração de dispositivos, é da opinião dos autores que o protocolo NETCONF pode ser eficientemente substituído pelo protocolo SOAP sem perda de funcionalidade e com ganho de desempenho em relação ao consumo de banda.

Como trabalho futuro, novas avaliações de NETCONF e SOAP para configuração serão verificadas. Em particular, novos encapsulamentos serão considerados, principalmente em relação à segurança (ex.: HTTPS e SSH) e a novos modelos de comunicação (ex.: BEEP). Além disso, outros parâmetros de desempenho serão verificados, como consumo de CPU e memória do dispositivo gerenciado.

Referências

- Arora, A. et al. (2004) “Web Services for Management (WS-Management)”, http://www.intel.com/technology/manage/downloads/ws_management.pdf, Outubro.
- Ayala, D. (2004) “nuSOAP, SOAP Toolkit for PHP”, <http://dietrich.ganx4.com/nusoap>, Agosto.
- Curbera, F. et al. (2002) “Unraveling the Web Services Web: an Introduction to SOAP, WSDL, and UDDI”, In: IEEE Internet Computing, Vol. 6, Issue 2, p. 86-93.
- DCOM (2004) “COM: Component Object Model Technologies”, <http://www.microsoft.com/com>, Outubro.
- Enns, R. (2004) “NETCONF Configuration Protocol”, IETF, *draft-ietf-netconf-prot-04*, Outubro (trabalho em andamento).
- Ethereal (2004) “Ethereal – The world’s most popular network protocol analyzer”, <http://www.ethereal.com>, Novembro.
- Goddard, T. (2004) “Using the Network Configuration Protocol (NETCONF) Over the Simple Object Access Protocol (SOAP)”, IETF, *draft-ietf-netconf-soap-03*, Setembro (trabalho em andamento).
- Lear, E. e Crozier, K. (2004) “Using the NETCONF Protocol over Blocks Extensible Exchange Protocol (BEEP)”, IETF, *draft-ietf-netconf-beep-03*, Novembro (trabalho em andamento).
- MacFaden, M. et al. (2003) “Configuring Networks and Devices with Simple Network Management Protocol (SNMP)”, IETF, RFC 3512, Abril.
- MiniXML (2004) “miniXML”, <http://minixml.psychogenic.com>, Outubro.
- Mitra, N. (2003) “SOAP Version 1.2 Part 0: Primer”, World Wide Web Consortium, <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>, Setembro.
- Neisse, R. et al. (2004) “Implementation and Bandwidth Consumption Evaluation of SNMP to Web Services Gateways”, In: 9th IFIP/IEEE Network Operations and Management Symposium (NOMS 2004), p. 715-728.
- Oasis Consortium (2004) “OASIS Consortium – Advancing E-Business Standards Since 1993”, <http://www.oasis-open.org/>, Outubro.
- Pear (2004) “PEAR :: The PHP Extension and Application Repository”, PHP Group, <http://pear.php.net>, Outubro.
- PHP (2004) “PHP: Hypertext Preprocessor”, The PHP Group, <http://php.net>, Julho.
- Raj, G. S. (2004) “CORBA - Common Object Request Broker Architecture”, <http://my.execpc.com/~gopalan/corba/corba.html>, Outubro.
- Wasserman, M. e Goddard, T. (2004) “Using the NETCONF Configuration Protocol over Blocks Secure Shell (SSH)”, IETF, *draft-ietf-netconf-ssh-02*, Outubro (trabalho em andamento).
- YENCA (2004) “Yenca Project”, OSTG - Open Source Technology Group, <http://sourceforge.net/projects/yenca/>, Agosto.