

Perfsonar: uma infraestrutura para monitoramento da qualidade de redes de computadores utilizando a internet**Perfsonar: an infrastructure for quality monitoring of computer networks over the internet**

Recebimento dos originais: 15/02/2019

Aceitação para publicação: 11/03/2019

Priscila da Silva Alves

Engenheira de Telecomunicações pela Universidade Federal do Rio Grande do Norte
Instituição: Universidade Federal do Rio Grande do Norte
Endereço: Rua Jaguarari 1745, Apt. 104, CEP 59.032-620, Bairro Lagoa Seca, Natal-RN,
Brasil
E-mail: priscila.contato@live.com

Gutemberg Soares da Silva

Doutor em Engenharia Elétrica e Computação pela Universidade Federal do Rio Grande do Norte
Instituição: Universidade Federal do Rio Grande do Norte
Endereço: Rua Miguel Barra 800, CEP 59.014-590. Apt. 1200, Bairro Tirol, Natal- RN,
Brasil
E-mail: guttembbergue@gmail.com.br

RESUMO

O perfSONAR é um sistema que funciona em ambiente linux e permite criar uma infraestrutura de monitoramento para redes de computadores utilizando a Internet. Ele realiza medições de vazão e latência através dos softwares BWCTL, NDT e OWAMP. Nesta pesquisa foi utilizado o perfSONAR para realizar dois testes de vazão TCP. O primeiro entre a Universidade Federal do Rio Grande do Norte (UFRN) e o Centro Brasileiro de Pesquisas Físicas. O segundo entre a UFRN e a Universidade de Lehigh, nos Estados Unidos. Os testes trafegaram sobre os backbones da rede IPÊ, da Rede Clara e da Internet2, utilizando largura de banda de 10 Gbps. Observou-se a viabilidade de implantação do perfSONAR como infraestrutura para o monitoramento contínuo da qualidade de transmissão da rede. Os resultados das medições de vazão demonstram quais os horários com maior ou menor taxa de uso da rede pelos usuários e a oscilação do canal ao longo do período de duração dos testes.

Palavras-Chave: perfSONAR, BWCTL, iPerf, vazão.

ABSTRACT

PerfSONAR is a system that works in a Linux environment and allows you to create a monitoring infrastructure for computer networks using the Internet. It performs flow and latency measurements through BWCTL, NDT and OWAMP software. In this research, perfSONAR was used to perform two TCP flow tests. The first between the Federal University of Rio Grande do Norte (UFRN) and the Brazilian Center for Physical Research.

The second between UFRN and the University of Lehigh in the United States. The tests traveled on the backbones of the IPÊ network, Clara Network and Internet2, using bandwidth of 10 Gbps. The feasibility of implementing perfSONAR as an infrastructure for the continuous monitoring of transmission quality of the network was observed. The results of the flow measurements show the schedules with greater or lesser use of the network by the users and the oscillation of the channel over the duration of the tests.

Keywords: perfSONAR, BWCTL, iPerf, flow.

1 INTRODUÇÃO

Para uma gestão eficiente da engenharia de tráfego é fundamental conhecer os elementos que influenciam no comportamento do tráfego que é transmitido pelo *backbone*. Albuquerque (2013) [1] identifica elementos que são necessários à realização de uma boa engenharia de tráfego: banco de dados com informações da rede, topologia da rede, estado dos componentes da rede, avaliação da demanda e roteamento inteligente.

Segundo Albuquerque (2013), a qualidade de serviço em redes de computadores compreende não só a qualidade do *backbone*, mas também a qualidade da conexão perceptível para os usuários. Assim, um sistema que consiga medir a *osciosidade* da rede pode fornecer dados úteis para a administração da rede.

Medições da capacidade instantânea do canal e do estado do link revelam quais são os momentos osciosos da rede, podem ser utilizados para identificar links subdimensionados e para realizar otimizações nos equipamentos.

O perfSONAR (*Performance Service Oriented Network Monitoring Architecture*) [11] foi idealizado e desenvolvido através de uma parceria entre a rede GÉANT, a Internet 2, a Universidade de Indiana e a ESnet. O seu objetivo é fornecer uma infraestrutura de medições a nível global para ser aplicada, principalmente, em redes acadêmicas, de modo a permitir o monitoramento de conexões fim-a-fim utilizando a Internet.

As medições em camada quatro são realizadas através do envio de sucessivos pacotes da aplicação de origem até a porta da aplicação de destino. A velocidade máxima que se consegue obter demonstra a disponibilidade do canal para transporte de arquivos naquele momento. Esta análise pode ser feita empregando-se os protocolos TCP (*Transmission Control Protocol*) [15] e UDP (*User Datagram Protocol*) [12].

O procedimento de abertura e encerramento de sessão do protocolo TCP é baseado no protocolo RDT (*Reliable Data Transfer*). Eles utilizam verificação de erros de bits, através do *checksum*, para garantir que não houveram erros durante a transmissão dos bits do pacote. Também utiliza confirmação positiva (ACK) ou negativa (NAK) de entrega, para

que o remetente saiba se o destinatário recebeu as informações. Ao enviar um pacote o remetente inicia um *timmer* para esperar a mensagem de confirmação positiva ou negativa, se essa mensagem não chegar o remetente reenvia o pacote. Para otimizar o uso do canal de comunicação e não desperdiçar a banda o TCP utiliza o conceito de paralelismo, ele estabelece janelas de comprimento N e envia a quantidade de pacotes que couberem na janela, sem necessariamente precisar receber a confirmação de recebimento de cada pacote antes de enviar o próximo.

Todo esse procedimento utiliza uma quantidade de pacotes para controle maior que a utilizada pelo protocolo UDP, que apenas realiza verificação de erros de bits através do *checksum*. Assim, a quantidade de pacotes de controle trocados antes de iniciar a transferência dos dados será maior com o TCP.

Os resultados obtidos com o protocolo TCP demonstram a vazão máxima da rede durante os testes. Esta vazão nunca será equivalente a taxa de transmissão máxima permitida pelo canal, pois os bits de controle e sincronismo, presentes no cabeçalho do protocolo TCP, não são considerados durante o cálculo da vazão. Os resultados utilizando o protocolo UDP, devido a sua natureza não orientada a conexão, são normalmente utilizados para obter amostras de *jitter* da rede e para indicar possíveis problemas de congestionamento e perdas do canal.

Uma ferramenta muito conhecida que realiza medições utilizando os protocolos de transporte TCP e UDP é o iPerf [4]. Ele foi desenvolvido em C++ pela *Distributed Applications Support Team* (DAST) e pelo *National Laboratory for Applied Network Research* (NLNR), seus autores são Jon Dugan, Seth Elliott, Bruce A. Mah, Jeff Poskanzer, KaustubhPrabhu. O iPerf utiliza endereçamento com o protocolo IP em sua versão 4 [13] e o iPerf3 [5], que é a última atualização da ferramenta, também fornece suporte para o protocolo IP na versão 6 [17].

Existem softwares, como o BWCTL [2], que permitem implementar uma infraestrutura de controle para os testes realizados pelo iPerf. O BWCTL é um software que realiza a alocação de recursos e escalonamento do sistema para permitir a realização simultânea de vários testes com o iPerf ou o iPerf3. O OWAMP [8] funciona de forma semelhante ao BWCTL, porém seu objetivo é fornecer uma infraestrutura para controlar testes de latência.

A proposta do *perfSONAR* é diferente da proposta do BWCTL, pois ele tem como objetivo testar o desempenho de todo o caminho percorrido pelos pacotes que saem de um

servidor e chegam no outro, passando pela Internet. Além de permitir testes agendados que serão executados na frequência definida. Continuamente é possível verificar nos resultados das medições de desempenho da rede as oscilações do circuito e, assim, observar quais foram os períodos de pior e melhor desempenho. Os resultados obtidos permitem a localização de conexões de baixa qualidade e o diagnóstico de possíveis falhas da rede.

2 REFERENCIAL TEÓRICO

Os parâmetros para avaliação do desempenho de dispositivos interconectados em rede são definidos pela RFC 1242 [16]. Dentre estes parâmetros, é destacado o *throughput*, que representa um valor para medir a taxa máxima de transferência suportada pela rede. A RFC 1242 aconselha que medições de *throughput* aconteçam utilizando tamanhos de frame diferentes e em períodos espaçados e pré-definidos de tempo. Outros parâmetros para avaliação de desempenho de redes especificados por ela são a taxa de perda de frames e a latência.

A latência é definida pela RFC1242 como o intervalo de tempo entre a saída do primeiro bit pela porta de origem e a chegada do último bit na porta de destino. A referência também aborda que a variação da latência pode ser considerada um problema para alguns protocolos de rede e que o seu aumento pode resultar na diminuição do uso da rede.

A RFC 1242 define a taxa de perda de frames como o percentual de frames que é encaminhado pela rede e que não chega ao destino. Ela informa que a taxa de perda de frames é utilizada para medir a performance da rede e identificar se a rede está trabalhando em estado de *overload* (sobrecarga).

2.1 LATÊNCIA

Em comunicações analógicas e digitais, a latência representa o atraso sofrido pelo sinal ao longo do tempo. Em redes de computadores, ela representa o atraso sofrido pelos bits transmitidos da origem até o destino. A latência é um parâmetro que utiliza os relógios do servidor NTP dos dispositivos de origem e destino para calcular o tempo que um pacote leva para ser transmitido da porta de saída do dispositivo de origem até a porta de entrada do dispositivo de destino.

2.2 LARGURA DE BANDA

Largura de Banda, em sistemas de comunicação analógicos, é uma medida que representa o espaço reservado no espectro de frequências para o respectivo canal de comunicação. Ela é normalmente fornecida em Hertz e pode ser calculada através da diferença entre a frequência máxima e mínima do sinal no espectro.

Em sistemas de comunicações digitais, a largura de banda representa taxa de transmissão, ou taxa de símbolos, do canal de transmissão. Ela expressa a quantidade de símbolos por segundo que pode ser transmitida em determinado canal. Normalmente ela é expressa em bits por segundo, mas comercialmente as grandezas mais adotadas são Mega bits por segundo (Mbps) e Giga bits por segundo (Gbps).

2.3 THROUGHPUT

Throughput, ou vazão, em português, é um parâmetro medido em bits por segundo e que representa a quantidade de informação que pode ser transmitida no canal de comunicação em um intervalo de tempo. Sabe-se que um símbolo compreende não apenas bits úteis (informação), como também bits de controle e sincronismo, então podemos dizer que o conceito de *throughput* difere do conceito de taxa de transmissão, pois a taxa de transmissão é calculada considerando todos os bits transmitidos pelo canal. A vazão é calculada considerando-se somente a informação que será transmitida.

2.4 JITTER

O *Jitter* é calculado através do desvio padrão entre a variação do atraso entre pacotes enviados sucessivamente. Ele é uma medida que indica a qualidade do canal para transmissão de fluxos de áudio e vídeo de forma interativa no tempo.

2.5 JITTER

O Ping [18] é uma ferramenta de linha de comandos que utiliza pacotes ICMP (*Internet ControlMessageProtocol*) [14], da camada de rede, para calcular alguns parâmetros que podem ser utilizados para medir a conectividade de uma rede. Ele identifica se o dispositivo de destino está acessível pela rede, através do envio de pacotes ICMP, calcula o atraso de ida-e-volta, a quantidade de pacotes transmitidos pelo destino, o tamanho em bytes dos pacotes enviados, o maior tempo de resposta do destino, o menor tempo de resposta do destino e a média dos tempos de resposta.

2.6 IPERF

O iPerf [4] é uma ferramenta que permite realizar medições ativas para calcular a vazão máxima que pode ser atingida por um circuito. Ele pode ser utilizado com os protocolos TCP, UDP e SCTP; permitindo testar a vazão, o jitter e a perda de pacotes de uma comunicação, em uma ou duas direções. Para utilizar o iPerf devemos, primeiramente, instalá-lo via apt-get nos dois pontos de medição. O primeiro passo para começar a medição é iniciar o serviço do iPerf no servidor, ao fazer isso o computador que irá se comportar como servidor durante os testes abrirá as portas necessárias do computador e ficará esperando as solicitações de testes. Depois, envia-se uma solicitação da máquina do cliente para se inicializar o teste. Estes são exemplos dos comandos que podem ser executados no cliente e no servidor:

- Servidor: `iperf -s -P 5002 -i 1 -f m -T 1`;
- Cliente: `iperf -c -i 1 -f m -t 3600 -T 1`.

O “-i” é um parâmetro para informar o intervalo de tempo em que o retorno do teste será exibido na tela, o “-f” é um parâmetro que informa a unidade de exibição dos resultados (kbits, Mbits). O tempo de execução do teste, “-t”, é um parâmetro do cliente e deve ser informado em segundos. O “-T” informa o *time-to-live* (TTL), também em segundos. O “-P” é um parâmetro do servidor para informar a porta que ele utilizará para os testes. Enquanto o teste estiver rodando ele retornará os resultados na tela para o usuário. Por padrão o iPerf realiza um teste TCP entre o Cliente e o Servidor, para realizar um teste UDP é necessário adicionar o parâmetro “-u”.

2.7 TRACEROUTE

O *Traceroute* [18] é outra ferramenta de linha de comandos que utiliza pacotes ICMP para buscar o caminho que um pacote irá percorrer na rede, da origem até o destino. O parâmetro informa todos os endereços de roteadores percorridos no caminho, em ordem, até o endereço final. Também informa o tempo de resposta de cada equipamento. É um comando útil para descobrir pontos de falhas em enlaces de redes, pois quando o pacote chegar ao ponto de falha a ferramenta retornará um “*” para indicar que o equipamento não responde ou que esta rede bloqueia pacotes ICMP.

2.8 OWANP

O OWAMP (*One-way Active Measurement Protocol*) é um protocolo definido na RFC 4656 [19] e implementado pelo software de mesmo nome [8]. O software OWAMP realiza o monitoramento da latência, da perda de pacotes e verifica o funcionamento da rede no sentido cliente-servidor de um sistema de comunicação. Ele é implementado através dos protocolos OWAMP-Control e OWAMP-Test. O OWAMPControl é utilizado para iniciar sessões de teste, parar sessões de teste e buscar os seus resultados. O OWAMP-Test é utilizado para solicitar a troca de pacotes entre os dois nós de medições.

2.9 NDT

O NDT (*Network Diagnostic Tool*) [7] é uma ferramenta desenvolvida pela Internet2 e utilizada no diagnóstico de problemas de rede, como gargalos no enlace de comunicação e incompatibilidade entre o caminho de ida e vinda. Ele também requer a implementação do algoritmo de controle de congestionamento TCP Reno, não dando suporte ao TCP Cubic. O NDT é utilizado no *perfSONAR* para permitir a realização de testes de latência. Segundo informa na documentação do *perfSONAR* [9], como o algoritmo TCP Reno roda mais devagar que o algoritmo Cubic, sua ativação exigiria mais processamento durante testes de *throughput*, o que prejudica o seu funcionamento. Por isso é recomendado que o NDT rode em um servidor *perfSONAR* que esteja dedicado a testes de latência. Este trabalho recomenda que o administrador de redes utilize dois servidores *perfSONAR*, um para realização de testes periódicos de latência e outros para testes periódicos de *throughput*, para que um serviço não prejudique o funcionamento do outro.

2.10 BWCTL

O BWCTL (*Bandwidth Test Controller*) [2] é uma ferramenta utilizada para controlar os recursos do sistema utilizados durante medições utilizando os protocolos: *iPerf*, *iPerf3*, *Ping*, *Nuttcp*, *Traceroute*, *Tracepath* e OWAMP. Ele é implementado numa instancia cliente e numa instancia servidor, mas para iniciar a execução de uma medição só é necessário que um dos lados solicitem a realização do teste, isso é feito com a utilização de um servidor NTP que serve de sincronismo entre os dois dispositivos durante o teste. A execução de medições utilizando o BWCTL é realizada através de uma aplicação de terminal, utilizando os comandos:

- Iniciando o serviço: `/etc/init.d/bwctld start`;
- Iniciando o processo cliente: `# bwctl -c <ip_servidor>`;

- Iniciando o processo no servidor: # bwctl -s <ip_cliente>.

Os parâmetros “-s” e “-c” indicam se o dispositivo será servidor ou cliente, nesta ordem, durante as medições. Também podem ser adicionados os parâmetros ao teste, para definir as características do teste, o “-T” é utilizado para selecionar a ferramenta a ser utilizada nos testes (Ex: iperf, iperf3), o “-f” é utilizado para selecionar a unidade de medição da velocidade de transmissão, o “-t” é utilizado para selecionar o tempo de durabilidade das medições, em segundos, o “-i” é utilizado para selecionar a velocidade com que os retornos do teste serão exibidos em tela e o “-c” para selecionar o host que vai receber os dados.

2.11 perfSONAR

O *perfSONAR* é uma ferramenta aberta e que pode ser utilizada para realizar testes de desempenho através de um caminho na rede. A filosofia do software parte do princípio de que todo o caminho pelo qual o pacote irá passar pode influenciar no desempenho da transmissão. Então ele fornece uma interface que permite realizar testes entre servidores *perfSONAR* cadastrados no Domínio Global e que estejam em qualquer lugar no mundo. Assim, pode-se verificar como está o desempenho da rede entre duas universidades, por exemplo. O *perfSONAR* permite a adição de complementos [10] que adicionam outras funcionalidades à ferramenta, como, por exemplo, o *perfSONAR* UI e o *MaDDash* (*Monitoring and Debugging Dashboard*). O *perfSONAR* UI é uma aplicação web do *perfSONAR* para solicitação de testes sob demanda, através de ferramentas como: *ping*, *traceroute*, *OWAMP* e *BWCTL*. Ele tem como pré-requisitos a instalação da máquina virtual Java e do pacote *Tomcat 6*, para liberar o acesso à interface web. O *MaDDash* é uma ferramenta que permite a implementação de uma malha de medições em domínio local utilizando o *perfSONAR* como agente ativo das medições, configurando e enviando os testes para os dispositivos conectados a malha.

Na documentação do *perfSONAR* [11] é possível observar políticas de segurança para o seu uso. Tais medidas devem ser consideradas, pois o servidor está disponível em domínio público, para acesso por qualquer usuário, inclusive usuários mal-intencionados. Como medida de segurança, é recomendado desabilitar o acesso remoto. Caso seja necessário habilitar o acesso remoto, é recomendado em sua documentação a liberação do *ssh* junto a uma configuração de *jump host*, realizada através do *IPTables*. Essa configuração permite que o acesso remoto seja realizado, porém apenas através de um único host local, ou

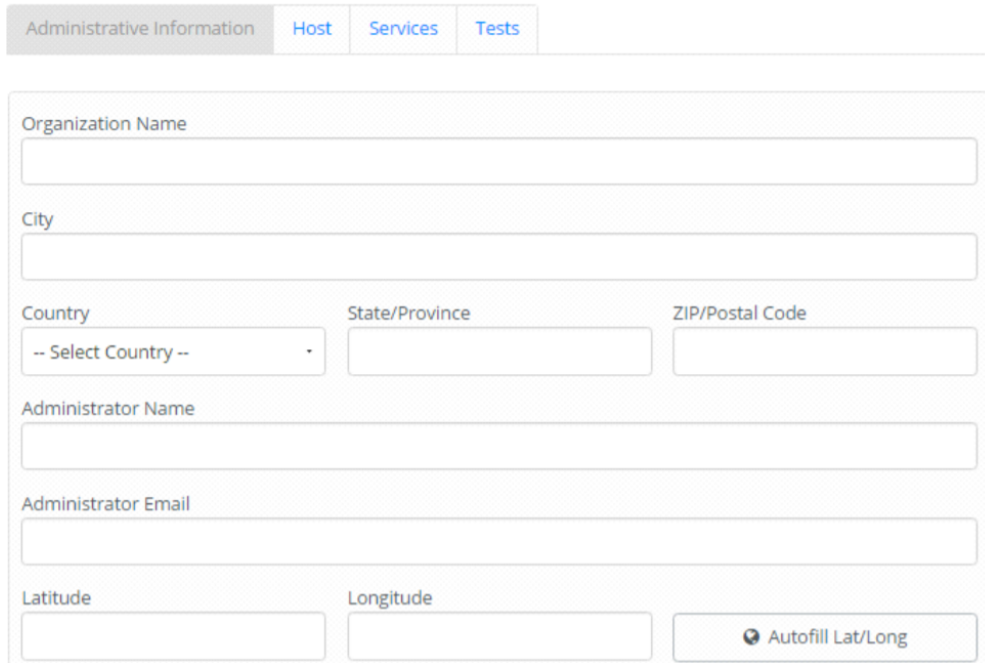
um conjunto de hosts, previamente designado. Também se pode implementar uma medida de segurança conhecida como SSHD *throttling*, onde o número de vezes que um mesmo host pode tentar realizar uma conexão é limitado. Assim, pode-se evitar ataques de força bruta.

3 MÉTODOS

Para Marconi e Lakatos (2003) [6], o método é o conjunto de atividades sistemáticas e racionais que, com maior segurança e economia, permite alcançar o objetivo de organizar os conhecimentos válidos sobre uma temática, traçar o caminho a ser seguido e auxiliando as decisões do pesquisador. Portanto, o método proposto nesta pesquisa consiste em estabelecer os procedimentos necessários para a realização de medições de *throughput* fim-a-fim. O *perfSONAR* provê uma interface web desenvolvida em *Django* que permite a configuração dos parâmetros envolvidos nos testes. Ela permite criar testes que serão executados e repetidos durante períodos pré-definidos de tempo. Assim, verifica-se o desempenho da rede de forma contínua. Simultaneamente à realização das medições o *perfSONAR* gera gráficos com os resultados e realiza testes de *traceroute*, para verificar por qual caminho do *backbone* os pacotes estão sendo transmitidos.

3.1 PRIMEIROS PASSOS

Durante a inicialização do servidor pela primeira vez o sistema solicita que você informe um usuário administrador e um usuário padrão para o sistema, e diga se quer habilitar ou não o acesso via ssh para esses dois usuários. Imediatamente fica disponível a interface web para acesso remoto do servidor, o login na interface web é utilizando um usuário administrador.



The image shows a web interface for configuring administrative information in perfSONAR. At the top, there are four tabs: 'Administrative Information' (selected), 'Host', 'Services', and 'Tests'. Below the tabs is a form with the following fields:

- Organization Name: A text input field.
- City: A text input field.
- Country: A dropdown menu with the text "-- Select Country --".
- State/Province: A text input field.
- ZIP/Postal Code: A text input field.
- Administrator Name: A text input field.
- Administrator Email: A text input field.
- Latitude: A text input field.
- Longitude: A text input field.
- Autofill Lat/Long: A button with a location pin icon and the text "Autofill Lat/Long".

Figura 1: vista da interface web do perfSONAR com informações administrativas

A primeira coisa que deve ser feita no servidor, a partir da interface web, é a configuração das informações administrativas, conforme mostra a Figura 1. Depois, na aba Hosts, são ativadas as atualizações automáticas e selecionados os servidores NTP utilizados para sincronização, conforme mostra a Figura 2. Na aba Services são selecionados os serviços que deverão ficar ativos no servidor, como pode ser visto na Figura 3.

Quando são adicionados ou removidos os servidores NTP listados na Figura 2 essas configurações ficam salvas nos arquivos “/etc/ntp.conf” e “/etc/ntp/step-tickers” do CentOS. Através do comando “ntpq -c pe”, no terminal do servidor, podemos verificar o delay e o jitter do serviço NTP para cada servidor configurado. Os comandos para verificar o horário, acompanhado do delay, e sincronizar os servidores são, respectivamente “hwclock --show” e “hwclock --systohc”.

A Figura 4 corresponde a interface de configuração de testes do perfSONAR. Os testes de Throughput e ping, que são utilizados durante a homologação dos circuitos, podem ser agendados com poucos clicks.

Administrative Information Host Services Tests

Auto Updates

Automatically updates all software on a nightly basis. [Click here](#) for important information regarding automatic updates.

Disabled

NTP Servers

Click or type below to configure your NTP servers.

- × chronos.es.net (ESnet - New York, NY USA)
- × owamp.chic.net.internet2.edu (Internet2 - Chicago, IL USA)
- × owamp.hous.net.internet2.edu (Internet2 - Houston, TX USA)
- × owamp.losa.net.internet2.edu (Internet2 - Los Angeles, CA USA)
- × owamp.newy.net.internet2.edu
- × saturn.es.net (ESnet - Sunnyvale, CA USA)

[Select the closest servers](#)
[Manage available NTP servers](#)

Figura 2: Configurações de sincronismo do servidor NTP

Administrative Information Host Services Tests

Select services you want enabled on this host

- BWCTL Allows clients at other sites to run Throughput, Traceroute and Latency tests to this host
- OWAMP Allows clients at other sites to run One-Way Latency tests to this host
- NDT Allows clients at other sites to run NDT tests to this host
- NPAD Allows clients at other sites to run NPAD tests to this host

[Select only bandwidth services](#) [Select only latency services](#)

Figura 3: Configurações dos serviços executados pelo servidor perfSONAR

The screenshot shows the 'Tests' tab in the perfSONAR configuration interface. At the top, there are navigation tabs: 'Administrative Information', 'Host', 'Services', and 'Tests'. Below the tabs, a green banner indicates 'Throughput tests will be running 39% of the time'. A configuration instruction reads 'Configure tests between this host and other hosts.' with '+ Host' and '+ Test' buttons. Below this, there is a 'View by: Test | Host' selector. The main content is a table of test configurations:

TEST NAME	TYPE	INTERVAL	TEST MEMBERS	ENABLED	ACTIONS
Teste geral 2	Throughput - TCP	2 hours	4 hosts	<input checked="" type="checkbox"/>	
perfSONAR Toolkit Default Traceroute Test	Traceroute	10 minutes	4 hosts	<input checked="" type="checkbox"/>	

Figura 4: Configuração de testes do servidor perfSONAR

3.2 MEDIÇÕES DE THROUGHPUT

Para iniciar um teste de *throughput* TCP é necessário selecionar a interface de rede que será utilizada para a saída e entrada da informação, o protocolo utilizado, o tempo entre os testes, o tempo de duração do teste e o valor utilizado pelo campo ToS (*Type of Service*) do protocolo IPv4 [13]. O ToS é um campo de 1 Byte de comprimento, que indica a prioridade que terá o pacote ao passar pelos roteadores. Ele adota o valor padrão 0 para indicar ausência de ToS, em roteadores que não implementam QoS (*Quality of Service*) o pacote será transmitido utilizando o valor padrão 0. Conforme observado no manual da Cisco [3] para atribuição de QoS em seus roteadores, o campo ToS é calculado utilizando a seguinte ordem:

- Precedência do IP: ocupa os três primeiros bits;
- *Delay, Throughput e Reliability*: ocupa os três bits do meio, após os bits de precedência;
- *Currently Unused*: ocupa os dois últimos bits.

Durante as medições o teste realizado utilizou o parâmetro de ToS selecionado em 0, pois desejava-se verificar a disponibilidade do canal de comunicação para o usuário comum,

então atribuir prioridade aos pacotes transmitidos durante as medições tornaria os resultados não condizentes com a real situação de usabilidade da rede.

A diferença entre um teste TCP e um teste UDP é que, no segundo, além de se selecionar os parâmetros descritos a cima, também deve ser fixada a largura de banda da rede. Isso ocorre pois, ao contrário do protocolo TCP, o UDP não realiza o controle de fluxo dos dados transmitidos. Assim, ele não consegue descobrir sozinho qual é a largura de banda utilizada pelo canal.

3.3 MEDIÇÕES DE LATÊNCIA

A configuração de testes de latência não deve ser realizada nos mesmos servidores em que se configura testes de *throughput*, pois a forma como eles utilizam o servidor NTP pode interferir na capacidade de processamento da máquina e, assim, prejudicar nos resultados das medições de vazão.

Por esta razão, escolheu-se não iniciar testes de latência no mesmo servidor utilizado para os testes de vazão. Para analisar o atraso do canal, são utilizados os resultados dos testes de *traceroute* que são executados para a realização dos testes de vazão.

3.4 DESCRIÇÃO DO EXPERIMENTO

Durante o experimento foram realizados dois testes de *throughput* TCP. O primeiro ocorreu entre a Universidade Federal do Rio Grande do Norte, identificado pelo IP 177.20.132.21, e o Centro Brasileiro de Pesquisas Físicas. Este teste foi executado dentro do *backbone* da Rede IPÊ [20], que é uma rede acadêmica nacional de fibra óptica e largura de banda de 10 Gbps. Os resultados podem ser observados na Tabela I.

O segundo teste foi executado entre a Universidade Federal do Rio Grande do Norte e a Universidade de Lehigh, localizada em Bethlehem, nos Estados Unidos. Este teste trafegou inicialmente pela Rede IPÊ, depois seguiu para a Rede Clara, composta por instituições de ensino e pesquisa da América Latina, para finalmente chegar na rede Internet2, também com capacidade de 10 Gbps.

A configuração de múltiplos testes de *throughput* em um mesmo servidor não irá interferir no funcionamento dos testes, pois o BWCTL realiza o gerenciamento dos recursos do sistema para que os testes não entrem em conflito. As tabelas a seguir mostram os parâmetros utilizados durante a configuração dos dois testes.

Tabela 1: Parâmetros definidos para o teste de vazão do cenário 1:

Tipo do teste	<i>Throughput</i>
Protocolo	TCP
Tempo entre os testes	2h
Tempo de duração do teste	10 segundos
ToS	0
Origem	177.20.132.21
Destino	ps02.cat.cbpf.br

Fonte: autoriapropriã.

Tabela 2: Parâmetros definidos para o teste de vazão do cenário 2:

Tipo do teste	<i>Throughput</i>
Protocolo	TCP
Tempo entre os testes	2h
Tempo de duração do teste	10 segundos
ToS	0
Origem	177.20.132.21
Destino	perfsonar.cc.lehigh.edu

Fonte: autoriapropriã.

4 ANÁLISE DOS DADOS E RESULTADOS OBTIDOS

As medições realizadas evidenciaram a oscilação do canal entre redes acadêmicas do Brasil, da América Latina e da América do Norte. Foram obtidos como resultados amostras de vazão fim-a-fim e do atraso exercido por cada um dos nós comutadores da rede. Pode-se verificar a capacidade máxima instantânea do canal e o tempo de transmissão total entre a origem e o destino para cada uma das redes.

Na medição 1, entre a UFRN e o CBPF, foi obtida uma vazão média igual a 333.61 Mbps e vazão reversa média igual a 369,16 Mbps. Os resultados do teste de vazão podem ser observados na Figura 5. Na Figura 6 é exibido o teste de *traceroute* que foi executado durante a execução do teste de vazão, para determinar a rota realizada pelos pacotes. Verificou-se que dois roteadores não responderam a solicitação de *traceroute*, ambos nas bordas das redes dos pontos de medição. Isso pode indicar medidas de segurança dos administradores de rede, para evitar descobertas de rede realizadas por pessoas mal-intencionadas. Os atrasos exercidos pelos nós de comutação estão disponíveis na Figura 6.

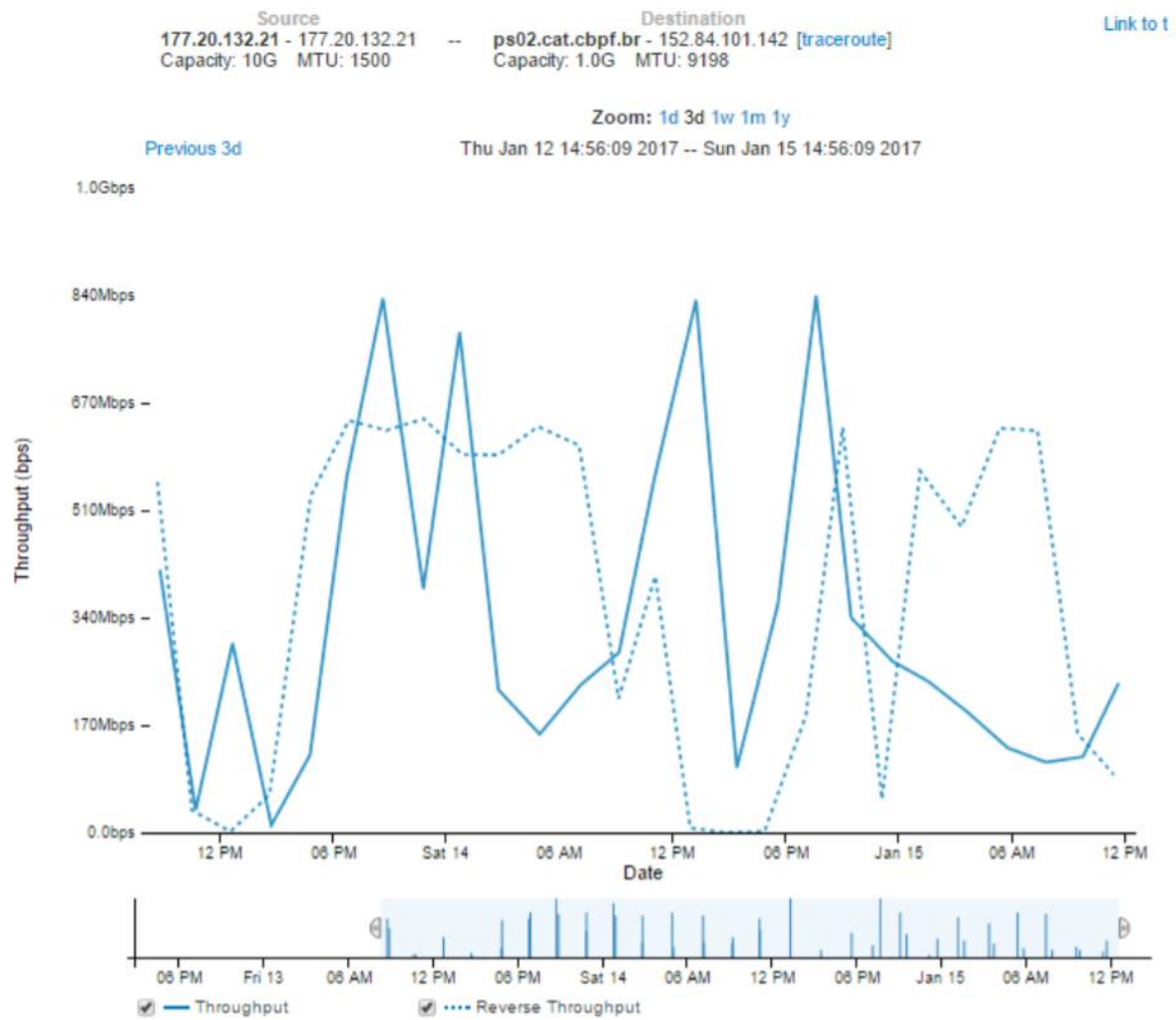


Figura 5: Resultado do teste de *throughput* e *throughput* reverso entre a UFRN e o CBPF.

Select endpoints available on <http://localhost/esmond/perfsonar/archive/>
 (177.20.132.21) ----> ps02.cat.cbpf.br (152.84.101.142) ▾
 Do not de-duplicate results

Topology beginning at Fri Jan 13 08:47:33 2017 (UTC -3)

Hop	Router	IP	Delay	MTU
1	177.20.132.1	177.20.132.1	0.194ms	1500
2	requestTimedOut	requestTimedOut		
3	sp-ufabc.bkb.rnp.br	200.143.255.61	0.634ms	1500
4	ce-rn-oi.bkb.rnp.br	200.143.252.138	8.085ms	1500
5	200.143.253.149	200.143.253.149	8.080ms	1500
6	sp-ce.bkb.rnp.br	200.143.253.26	50.899ms	1500
7	sp2-sp.bkb.rnp.br	200.143.253.38	51.182ms	1500
8	mg-sp2-oi.bkb.rnp.br	200.143.252.74	60.051ms	1500
9	df2-mg-oi.bkb.rnp.br	200.143.252.82	77.826ms	1500
10	rj-df2-oi.bkb.rnp.br	200.143.252.77	97.364ms	1500
11	rj-rederio.bkb.rnp.br	200.143.254.137	120.395ms	1500
12	200.20.96.4	200.20.96.4	97.500ms	1500
13	200.20.98.230	200.20.98.230	97.357ms	1500
14	requestTimedOut	requestTimedOut		
15	ps02.cat.cbpf.br	152.84.101.142	97.451ms	1500

Figura 6: Resultado de teste de *traceroute* entre a UFRN e o CBPF.

Na medição 2, entre a UFRN e a Universidade de Lehigh, obteve-se uma vazão média igual a 99,07 Mbps e uma vazão reversa média igual a 111.03 Mbps. Os resultados do teste de vazão podem ser observados na Figura 7. Na Figura 8 é exibido o teste de *traceroute* que foi executado durante a execução do teste de vazão. Desta vez, apenas o comutador de borda da UFRN não respondeu a solicitação de *traceroute*. Os atrasos exercidos pelos nós de comutação durante esse teste estão disponíveis na Figura 8.

Medições de vazão em redes de computadores podem fornecer diferentes resultados, dependendo da forma como foram realizadas. Se o canal de comunicação for reservado para o teste, a vazão demonstrará a capacidade de transmissão máxima conseguida no canal. Todavia, se o teste de vazão for executado em um canal que está sendo utilizado, os resultados demonstraram a oscilação do canal. Isso indica a taxa de transmissão instantânea que se consegue obter no link em questão, naquele momento de utilização da rede.

Os resultados obtidos demonstraram uma oscilação do canal de comunicação durante a realização dos testes, principalmente na medição 1. Isso pode indicar se tratar de um canal mal dimensionado ou com alta taxa de tráfego em horários específicos. Quando ocorrem interrupções na execução do teste, as informações são exibidas no gráfico através de erros em vermelho.

A principal aplicação dos resultados das medições de oscilações fim-a-fim em camada quatro é a análise dos melhores horários para transferência de grandes arquivos pela rede, o que é muito comum em redes acadêmicas. Com o perfSONAR, pode-se verificar quais links possuem melhores taxas de vazão na malha de medições, o que permite ao administrador da rede configurar a melhor rota entre a origem e o destino.



Figura 7: Resultado do teste de throughput e throughput reverso entre a UFRN, no Brasil, e a Universidade de Lehigh (USA).

As oscilações da rede e as taxas de vazão mais altas foram maiores nos resultados da medição 1, tanto para vazão direta como para vazão reversa. Ao analisar o caminho percorrido pelos pacotes referentes às duas medições, observa-se que até o sexto salto os dois testes realizaram o mesmo percurso. Ambos com atrasos médios muito próximos para

os seis primeiros comutadores. Porém, ao chegar no *backbone* da RNP de São Paulo os pacotes da medição 1 tiveram que passar por mais nove saltos até chegar no destino. Enquanto isso, os pacotes da medição 2 só precisaram passar por mais cinco saltos até chegar ao seu destino. Embora os atrasos da medição 2 sejam maiores que os da medição 1, devido à grande distância geográfica, o fato dos pacotes passarem por menos saltos influencia no tempo total de transmissão. Pode-se estimar que, durante os testes, o *backbone* da RNP de São Paulo estivesse passando por momentos de congestionamento e que, por isso, os pacotes tiveram que percorrer muitos comutadores até chegar ao destino.

Os pontos em vermelho no gráfico da Figura 7 mostraram erros de *throughput* reverso, que ocorreram devido a interrupção do teste do BWCTL por parte do cliente. Isso demonstra que o perfSONAR não só realiza testes de *throughput*, como também informa sobre as condições de realização do teste, alertando ao administrador do sistema sobre condições que interfiram na realização do teste e que podem ter influenciado nos resultados obtidos. Pode-se observar, por análise gráfica, que apesar de ocorrerem erros os testes de *throughput* continuaram sendo executados, então isso indica que eles foram interrompidos durante sua execução, e não foram bloqueados antes de iniciar. Ou seja, o cliente não bloqueou as suas portas para receber solicitações de testes de *throughput* reverso, mas, por algum motivo, os testes foram interrompidos. Isso é evidenciado no gráfico, pois inicialmente a vazão reversa era próxima a 270 Mbps e ao iniciarem os erros ela caiu praticamente pela metade. Um fator que pode ter desencadeado os erros do teste de vazão reversa é a reinicialização do processo do BWCTL/iPerf3 no cliente.

Select endpoints available on <http://localhost/esmond/perfsonar/archive/>
 (177.20.132.21) ----> perfsonar.cc.lehigh.edu (128.180.38.2) ▾
 Do not de-duplicate results

Topology beginning at Fri Jan 13 08:48:44 2017 (UTC -3)

Hop	Router	IP	Delay	MTU
1	177.20.132.1	177.20.132.1	0.201ms	1500
2	requestTimedOut	requestTimedOut		
3	sp-ufabc.bkb.rnp.br	200.143.255.61	0.558ms	1500
4	ce-rn-oi.bkb.rnp.br	200.143.252.138	8.032ms	1500
5	200.143.253.149	200.143.253.149	8.153ms	1500
6	sp-ce.bkb.rnp.br	200.143.253.26	50.830ms	1500
7	et-3-3-0.469.rts.wash.net.internet2.edu	64.57.28.61	183.995ms	1500
8	204.238.76.33	204.238.76.33	187.641ms	1500
9	204.238.76.46	204.238.76.46	187.555ms	1500
10	162.223.17.134	162.223.17.134	190.970ms	1500
11	perfsonar.cc.lehigh.edu	128.180.38.2	189.944ms	1500

Figura 8: Resultado de teste de *traceroute* entre a UFRN, no Brasil, e a Universidade de Lehigh (USA).

5 CONCLUSÕES

O *perfSONAR* é um servidor que realiza medições de vazão, atraso em uma direção, *ping* e *traceroute*. As medições realizadas pela interface web podem ser agendadas, com o tempo de duração e o tempo entre o intervalo das medições definido pelo administrador do sistema. O uso do *perfSONAR* como ferramenta para medir as oscilações em redes de computadores, através da Internet, é interessante pois permite ao administrador da rede ter resultados do desempenho instantâneo de conexões fim-a-fim entre os vários caminhos que os pacotes podem percorrer pela rede.

Durante a pesquisa foram analisados dois cenários, realizados testes de vazão, atraso em uma direção e *traceroute* para cada um dos cenários. Os pontos de medição do cenário 1 foram a Universidade Federal do Rio Grande do Norte e o Centro Brasileiro de Pesquisas Físicas, conectadas através Rede IPÊ, que é administrada pela Rede Nacional de Ensino e Pesquisa. Os pontos de medição do cenário 2 foram a Universidade Federal do Rio Grande do Norte e a Universidade de Lehigh, conectados através de três redes acadêmicas: a Rede IPÊ, a Rede Clara e a Internet2. Ambos os cenários possuíam largura de banda de 10 Gbps.

Os resultados mostraram que a utilização do *perfSONAR* foi adequada para realizar testes de desempenho através de um caminho de rede, contemplando medições de vazão, atraso em uma direção, *ping* e *traceroute*. Foi possível observar os horários de maior e menor utilização da rede, isso é notado através dos horários de medição de vazão onde se

conseguiu ter menores e maiores taxas de vazão. Com estes resultados é possível analisar o estado do link fim-a-fim entre os dois pontos de medição (vazão média conseguida pelo canal ao longo do tempo), o estado dos componentes da rede (atraso médio adicionado por cada comutador) e avaliar a demanda de tráfego no link em horários diferentes ao longo do dia.

Verificou-se que o *perfSONAR*, através de seu sistema de medições, permite identificar pontos problemáticos da rede, como links subdimensionados e que fornecem taxas de vazão muito baixas ou atraso muito alto, e estimar a qualidade da conexão perceptível para os usuários.

REFERÊNCIAS

- [1] ALBUQUERQUE, Edison. *QoS: Qualidade de Serviço em Redes de Computadores*. Editora Campus, 2013.
- [2] BWCTL. *Bandwidth Test Controller (BWCTL)*. Acessado em 1 de janeiro de 2015. Disponível em: <http://software.internet2.edu/bwctl/>.
- [3] CISCO. *Implementing Quality of Service Policies with DSCP*. Acessado em 1 de janeiro de 2015. Disponível em: [http://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-packet-marking/10103-dscpvalues.html](http://www.cisco.com/c/en/us/support/docs/quality-of-service/qos/qos-packet-marking/10103-dscpvalues.html).
- [4] iPerf. *iPerf - The ultimate speed test tool for TCP, UDP and SCTP*. Acessado em 1 de janeiro de 2015. Disponível em: www.iperf.fr.
- [5] iPerf3. *Change between iPerf 2.0, iPerf 3.0 and iPerf 3.1*. Acessado em 1 de janeiro de 2015. Disponível em: <https://iperf.fr/iperf-doc.php>.
- [6] MARCONI, M. A. LAKATOS, E. M. *Fundamentos da metodologia científica*. 5. ed. São Paulo: Atlas, 2003.
- [7] NDT. *Network Diagnostic Tool (NDT)*. Acessado em 1 de janeiro de 2015. Disponível em: <http://software.internet2.edu/ndt/>.
- [8] OWAMP. *One-Way Ping (OWAMP)*. Acessado em 1 de janeiro de 2015. Disponível em: <http://software.internet2.edu/owamp/>

[9] perfSONAR. *Network Diagnostic Tool (NDT)*. Acessado em 1 de janeiro de 2015. Disponível em: http://docs.perfsonar.net/using_ndt.html.

[10] perfSONAR. *Installing Additional Tools*. Acessado em 1 de janeiro de 2015. Disponível em: http://docs.perfsonar.net/manage_extra_tools.html.

[11] perfSONAR. Acessado em 1 de janeiro de 2015. Disponível em: www.perfsonar.net.

[12] RFC 768. *User Datagram Protocol*. Acessado em 1 de janeiro de 2015. Disponível em: <https://www.ietf.org/rfc/rfc768.txt>.

[13] RFC 791. *Internet Protocol*. Acessado em 1 de janeiro de 2015. Disponível em: <https://tools.ietf.org/html/rfc791>.

[14] RFC 792. *Internet Control Message Protocol*. Acessado em 1 de janeiro de 2015. Disponível em: <https://tools.ietf.org/html/rfc792>

[15] RFC 793. *Transmission Control Protocol*. Acessado em 1 de janeiro de 2015. Disponível em: <https://tools.ietf.org/html/rfc793>.

[16] RFC 1242. *Benchmarking Terminology for Network Interconnection Devices*. Acessado em 1 de janeiro de 2015. Disponível em: <https://www.ietf.org/rfc/rfc1242.txt>.

[17] RFC 2460. *Internet Protocol Version 6 (IPv6) Specification*. Acessado em 1 de janeiro de 2015. Disponível em: <https://www.ietf.org/rfc/rfc2460.txt>.

[18] RFC 2925. *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*. Acessado em 1 de janeiro de 2015. Disponível em: <https://www.ietf.org/rfc/rfc2925.txt>

[19] RFC 4656. *A One-way Active Measurement Protocol (OWAMP)*. Acessado em 1 de janeiro de 2015. Disponível em: <https://tools.ietf.org/html/rfc4656>.

[20] RNP. *Rede IPÊ*. Acessado em 1 de janeiro de 2015. Disponível em: <https://www.rnp.br/servicos/conectividade/rede-ipe>.