UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
ENGENHARIA DE COMPUTAÇÃO

MARCUS VINICIUS SOLDERA GRANDO

Avaliação de desempenho do protocolo OWAMP e medidas comparativas com protocolos *two-way*

Prof. Dr. Sérgio Luis Cechin
Orientador

Porto Alegre, Novembro de 2010

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, pois é Ele que nos dá a vida e a força para transpor os obstáculos.

Agradeço ao meus pais, Sonia e Álvaro, que acreditam que a educação é o maior patrimônio que podem me deixar, que sempre torceram por mim e me deram todo o suporte necessário para mais esta realização pessoal.

Meus agradecimentos especias à minha esposa Luciana, que durante praticamente toda a faculdade foi minha namorada, por estar do meu lado e ter me proporcionado dividir todos momentos, bons e ruins, com ela.

Agradeço também aos meus amigos e colegas que contribuíram para a realização de vários trabalhos ao longo da graduação e que contribuíram para a formação da pessoa que sou hoje. Acrescento aqui também meu irmão Mauricio, meu primo Allan e meu amigo Germano que dividiram uma casa comigo e sempre foram "parceria"e também a toda minha família que embora longe sempre esteve presente na minha vida.

Não poderia deixar de agradecer especialmente ao professor Dr. Sérgio Luis Cechin por ter me orientado na execução deste trabalho, sempre pronto para ajudar, chamar a atenção e contribuir tanto para esse trabalho como para o aumento da qualidade do curso de Engenharia de Computação da Universidade Federal do Rio Grande do Sul.

Dirijo também, meu reconhecimento à Universidade Federal do Rio Grande do Sul e ao Instituto de Informática por oferecer um curso de qualidade inquestionável e todos os equipamentos, condições e materiais necessários, sempre.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS 5

LISTA DE FIGURAS 6

LISTA DE TABELAS 7

RESUMO 8

ABSTRACT 9

1 INTRODUÇÃO 10
1.1 Objetivos 10

2 PESQUISA BIBLIOGRÁFICA 12
2.1 Modelo MR-OSI 12
2.1.1 Protocolo UDP 12
2.1.2 Protocolo TCP 13
2.2 Políticas de gerenciamento de despacho de pacotes no Linux 13
2.3 Atraso 14
2.4 A importância do tempo 14
2.4.1 Propriedades dos relógios 15
2.5 Network Time Protocol 16
2.5.1 A topologia do NTP 17
2.6 Medidas *one-way* e *two-way* 17
2.7 One-way Active Measurement Protocol 18
2.7.1 OWAMP-Control 19
2.7.2 OWAMP-Test 19
2.7.3 Implementações disponíveis 22

3 ESPECIFICAÇÃO DOS TESTES 23
3.1 Descrição dos testes 23
3.2 Equipamento utilizado 24
3.3 Cenário de testes 24
3.4 Configuração das máquinas e ferramentas utilizadas 25

4 EXECUÇÃO E RESULTADOS DOS TESTES 27
4.1 Testes IPv4 27
4.2 Testes IPv6 28
4.3 OWAMP versus J-OWAMP 29

4.4 OWAMP versus PING 31
5 CONCLUSÃO 33
REFERÊNCIAS 34

LISTA DE ABREVIATURAS E SIGLAS

FIFO	First In First Out
GPS	Global Positioning System
ISO	International Organization for Standarization
MR-OSI	Open System Interconnection Reference Model
NETEM	Network Emulator
NTP	Network Time Protocol
QoS	Quality of Service
RED	Random Early Detection
RTT	Round Trip Time
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

LISTA DE FIGURAS

Figura 2.1: Atraso médio versus capacidade oferecida 14
Figura 2.2: Topologia do NTP 17
Figura 2.3: Protocolos *two-way* 18
Figura 2.4: Topologia do protocolo OWAMP 19
Figura 2.5: Pacote no modo sem autenticação 20
Figura 2.6: Pacote no modo com autenticação ou criptografia 21
Figura 2.7: Detalhes do campo *timestamp* 21
Figura 2.8: Detalhes do campo estimativa de erro 21

Figura 3.1: Cenário de teste 25

Figura 4.1: Atrasos IPv4 - Descrição do teste *versus* Tempo(ms) 27
Figura 4.2: Atrasos IPv6 - Descrição do teste *versus* Tempo(ms) 29
Figura 4.3: Atrasos J-OWAMP *versus* OWAMP - Descrição do teste *versus* Tempo(ms) 30
Figura 4.4: Atrasos PING *versus* OWAMP - Descrição do teste *versus* Tempo(ms) 32

LISTA DE TABELAS

Tabela 2.1: As 7 camadas do modelo MR-OSI 12

Tabela 4.1: Resultados dos testes IPv4 28
Tabela 4.2: Resultados dos testes IPv6 28
Tabela 4.3: Resultados dos testes J-OWAMP *versus* OWAMP 30
Tabela 4.4: Resultados dos testes PING 31

RESUMO

Este trabalho apresenta um estudo do protocolo OWAMP para medições de tempos *one-way*. Uma visão geral das medições de tempo, *one-way* e *two-way*, é dada, explicando suas diferenças, vantagens e limitações. É efetuado um teste funcional de uma implementação do protocolo OWAMP. Também é feita a comparação das duas implementações existes do protocolo OWAMP a fim de evidenciar suas precisões e interoperabilidade. Por fim é executado um comparativo em uma das implementações OWAMP com o programa PING a fim de evidenciar as diferenças e aferir precisão de medidas *one-way versus two-way.*

Palavras-chave: OWAMP, *One-Way*, NTP, QoS, Desempenho de redes.

OWAMP protocol performance evaluation with two-way protocols comparative measures

ABSTRACT

This work presents a study of the OWAMP protocol for one-way time measurement. An overview of time measure, one-way and two-way, is given, explaining the differences, advantages and limitations of each one. A functional test of one OWAMP protocol implementation is made. Then a comparison between the two OWAMP protocol implementations is made to evaluate precision and interoperability. Finally, a comparative test is executed between one of the OWAMP implementations and the PING program to show the differences and check precision of one-way and two-way measures.

Keywords:

.

1 INTRODUÇÃO

Uma rede de computadores é um conjunto de computadores conectados entre si. Desde o início do desenvolvimento das redes de computadores estas vêm ganhando popularidade e os usos são os mais diversos possíveis. Desde troca de mensagem entre amigos, passando por redes de pesquisa, *grids* de computadores, redes de controle de processos nas indústrias, domótica, entre outros. Cada vez mais as redes se tornam importantes na vida das pessoas num mundo que está a cada dia mais interconectado.

Para falar de redes é necessário citar protocolos e meios de comunicação. As regras para a representação de dados e sinalização e o canal por onde é feita essa comunicação, respectivamente. Existe um modelo de referência para interconexões e protocolos, é o chamado MR-OSI (*Open System Interconection Reference Model*), que é uma descrição abstrata para o projeto de protocolos de rede que atuam sobre um canal de comunicação.

As redes de computadores são baseadas no MR-OSI, o qual divide a arquitetura da rede em sete camadas que devem ser independentes de maneira que possam serem modificadas para atender novos requisitos.

Devido ao fato de mais e mais pessoas terem acesso à rede mundial de computadores é inegável que a demanda por esses recursos só aumenta. Desta maneira é preciso gerenciar os recursos de rede de modo a prover um serviço de qualidade superior sem abrir mão da rentabilidade.

Para os administradores de rede é imprescindível poder reconhecer caminhos críticos, ou seja, locais onde congestionamentos possam estar acontecendo e contribuindo para a lentidão de uma rota ou da rede como um todo. Devido as proporções que as redes estão se tornando e com protocolos de roteamento cada vez mais complexos é comum acontecerem casos de roteamento assimétrico, ou seja, o caminho para alcançar determinado nodo não é necessariamente o caminho para retornar ao ponto de origem.

Neste cenário a utilização de ferramentas de medidas de *round trip time* tendem perder espaço, pois não apresentam um reflexo real do comportamento de uma rede onde aconteça roteamento assimétrico. Para isso existe o protocolo OWAMP (*One-Way Active Measurement Protocol*), capaz de determinar medidas de tempo unidirecionais, ou seja, podem determinar o tempo gasto para alcançar um determinado nodo da rede a partir de outro e a rota contrária separadamente. Mostrando dessa forma o comportamento real que um tráfego assimétrico pode gerar.

1.1 Objetivos

O objetivo deste trabalho é estudar e testar as implementações do protocolo OWAMP a fim de desvendar se são uma alternativa viável para execução de medidas *one-way*. Serão testadas duas implementações: Internet2 e J-OWAMP. Os testes serão efetuados em di-

ferentes configurações de modo de operação e diferentes procolos IP (*Internet Protocol*). Os modos de operação serão: aberto, com autenticação e com criptografia. Os protocolos IP utilizados serão o IPv4 e o IPv6. Para os testes de funcionalidade será utilizado o programa OWAMP da Internet2. Será apresentada uma comparação entre os dois programas OWAMP existentes. Haverá também uma comparação do programa OWAMP (Internet2) com o programa PING a fim de evidenciar as diferenças de medidas *one-way* e *two-way*.

2 PESQUISA BIBLIOGRÁFICA

2.1 Modelo MR-OSI

Com a crescente utilização de computadores ligados em rede em meados da década de 70 se viu necessária uma organização dessas redes. Por isso a ISO (*International Organization for Standardization*) criou um comitê técnico para elaboração de um padrão de comunicação capaz de ser interoperável entre os diversos sistemas (CARISSIMI, 2009, pág 62). No ano de 1983 foi então acolhido pela ISO o padrão MR-OSI sob a sigla ISO 7498.

Este padrão é baseados em camadas, descristas na tabela 2.1, que são independentes umas das outras. Isto é, cada camada fornece um serviço para a camada imediatamente superior e uma camada não pode alterar dados de outras camadas.

Nível	Nome
Nível 1	Nível físico
Nível 2	Nível de enlace
Nível 3	Nível de rede
Nível 4	Nível de transporte
Nível 5	Nível de sessão
Nível 6	Nível de apresentação
Nível 7	Nível de aplicação

Tabela 2.1: As 7 camadas do modelo MR-OSI

As descrições detalhada de cada camada serão omitidas neste trabalho, apenas é salientado que os pacotes utilizados pelo protocolo OWAMP estão descritos por protocolos da camada de nível 4, transporte, sendo eles o UDP e o TCP descritos a seguir nas seções 2.1.1 e 2.1.2.

2.1.1 Protocolo UDP

O protocolo UDP (*user datagram protocol*) é um protocolo da camada de transporte extremamente simples. É um protocolo não orientado à conexão e sem controle de perdas, duplicação ou ordenação de mensagens. O UDP fornece um mecanismo para multiplexação sobre o IP que inclui a confiabilidade da integridade dos dados através de *checksums* (CARISSIMI, 2009, pág 287).

O cabeçalho UDP é composto de 4 campos de 16 *bits* cada, sendo eles: porta de origem, porta de destino, tamanho do pacote e *checksum*. Os campos porta de origem

e porta de destino contém informações sobre qual processo deve receber o pacote na origem e no destino, respectivamente. No campo tamanho do pacote está o total em *bytes*, incluíndo cabeçalho UDP e dados do usuário, do pacote. Como o cabeçalho UDP é de tamanho fixo de 8 *bytes* este campo deverá conter no mínimo o valor 8.

O campo *checksum* contém dados para validar a integridade do pacote transmitido. Para o cálculo deste é levado em consideração o cabeçalho do UDP, os dados do usuário e um pseudo cabeçalho IP (endereço IP de origem, endereço IP de destino, tamanho do datagrama, 1 *byte* identificador do protocolo e 1 *byte* em zero). Está prática fere uma das recomendações da OSI que é a transparência entre os protocolos.

Por ser um protocolo razoavelmente leve o UDP é utilizado nas seções de teste do OWAMP, como descrito a seguir na seção 2.7.2.

2.1.2 Protocolo TCP

O TCP (*transmission control protocol*) é um protocolo que fornece um serviço de transporte, sobre IP, orientado à conexão (CARISSIMI, 2009, pág 291). Além disso o serviço é confiável no sentido de que pacotes perdidos na rede são retransmitidos, dados duplicados são descartado e que fornece os dados ao processo destino de forma ordenada, após reordená-los na máquina destino se necessário. Como o UDP, os processos de origem e destino são endereçados através do uso de portas.

O cabeçalho TCP é mais complexo que o do UDP, além de conter informações acerca das portas de origem e destino, tamanho do pacote e *checksum* existem campos para o controle de sequência e recebimento, alguns bits de sinalização e alguns outro dados relevantes ao funcionamento do protocolo. O tamanho mínimo do cabeçalho é de 20 *bytes* podendo ser maior devido aos campos opcionais existentes.

A parte importante do TCP para esse trabalho é que ele é confiável, por isso é utilizado na seção do protocolo OWAMP responsável pelo controle das seções de teste, como descrito a seguir na seção 2.7.1.

2.2 Políticas de gerenciamento de despacho de pacotes no Linux

Em um roteador é necessário que existam políticas de gerenciamento de despacho de pacotes, ou seja, é preciso existir algum mecanismo que determine como os pacotes recebidos na interface de entrada são encaminhados a respectiva interface de saída. Estas políticas são denominadas de políticas de gerenciamento de despacho de pacotes. Estas disciplinas são importantes como ferramentas para prover QoS *Quality of Service*, em contrapartida protocolos como o OWAMP são ferramentas para verificar se as políticas de QoS estão sendo efetivas.

Em roteadores baseados em Linux existem diversas políticas que implementam essas funcionalidades, entre elas:

TBF *Token Bucket Filter*

FIFO *First In First Out*

NETEM *Network Emulator*

RED *Random Early Detection*

O TBF serve para limitar a vazão de determinado fluxo, filas FIFO são as que simplesmente reenviam primeiro os pacotes que chegarem primeiro, NETEM é uma disciplina de

gerenciamento para emular redes e a RED leva em conta os mecanismos de adaptação de vazão do TCP para evitar possíveis congestionamentos. Neste trabalho a disciplina NETEM será utilizada para a execução dos testes.

2.3 Atraso

Todo tráfego de comunicação está sujeito à atrasos, os quais geralmente se anseia que sejam reduzidos ao mínimo possível. Nos casos mais simples o atraso consiste na quantidade de tempo necessário para transmitir um tráfego qualquer (HIGGINBOTTOM, 1998, pág 5). Em vários casos existem fontes adicionais de atraso, somando-se ao atraso de transmissão, que contribuem significativamente com o atraso total. Na realidade o atraso de transmissão geralmente é insignificante se comparado ao tempo de configuração da conexão ou o tempo gasto em filas, por exemplo.

O atraso médio, ou tempo de espera, é uma das medidas tradicionais em análise de tráfego de telecomunicações que são relevantes para roteadores equipados com disciplinas de fila. O atraso médio é também plotado como uma função da intensidade de tráfego, vazão, e carga disponível para fornecer uma característica de desempenho que é útil para investigar a eficiência de protocolos de acesso a LAN e redes de alta velocidade. Um gráfico típico de atraso médio versus vazão está plotado na figura 2.1, onde o atraso está em unidades de tempo e a vazão está normalizada. Usualmente as unidades de atraso são normalizadas de acordo com o tempo de transmissão de um pacote e, como na figura 2.1, a vazão é normalizada de acordo com a capacidade ideal. Notar que a capacidade real do sistema em estudo está indicado pela assíntota da curva.

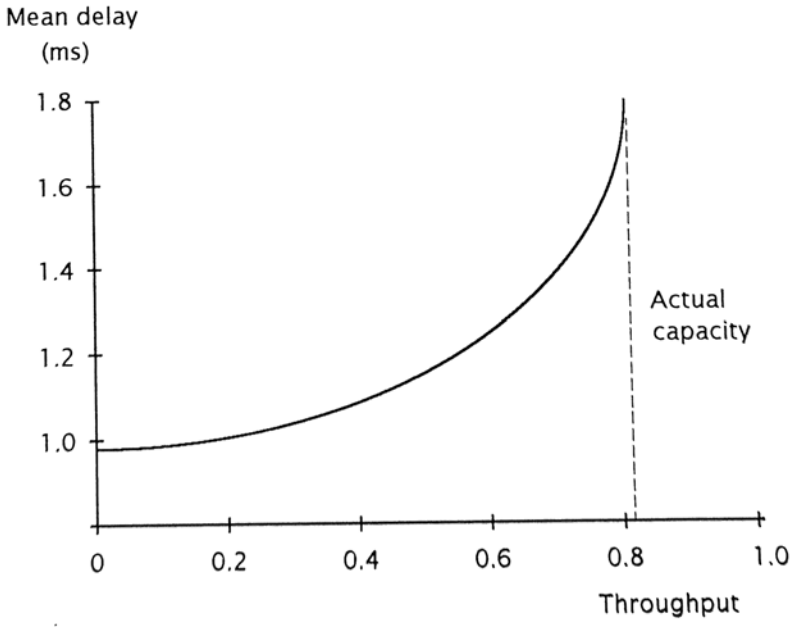


Figura 2.1: Atraso médio versus capacidade oferecida

2.4 A importância do tempo

Algumas aplicações podem ser sensíveis a problemas relativos à sincronização do tempo. Alguns dos casos em que podem levar estas aplicações à erros são descritos a

seguir:

- Um ou mais computadores estarem com o tempo diferente da hora legal;
- Um computador com o tempo no passado;
- Computadores discordando entre si quanto ao tempo correto.

A seguir alguns exemplos de aplicações onde é desejável que o tempo esteja ajustado com a hora legal:

- Agendadores de eventos;
- Aplicações de tempo real;
- Criptografia;
- **Protocolos de comunicação**;
- Sistemas de arquivos;
- Sistemas de backup remoto;
- Sistemas de distribuição de conteúdo;
- Sistemas transacionais e bancos de dados distribuídos.

Na lista acima encontra-se em negrito **"Protocolos de comunicação"** por ser exatamente o caso do OWAMP, objeto de estudo deste trabalho. No caso particular do OWAMP além do relógio estar sincronizado ele precisa estar muito preciso devido a magnitude dos valores medidos pelo protocolo.

Enfim nota-se que a sincronização do relógio, seja ela da ordem de segundos ou milissegundos, é importante para diferentes aplicações. Principalmente quando se conecta um computador à uma rede com diversos outros computadores.

2.4.1 Propriedades dos relógios

O relógio dos computadores são baseados em 3 elementos: um oscilador, um contador e um dispositivo de leitura. O oscilador é um dispositivo que gera eventos recorrentes numa frequência constante, geralmente nos computadores são cristais de quartzo. O contador acumula os ciclos gerados pelo oscilador e converte em unidades de medidas conhecidas (segundos, milissegundos, etc) que são chamadas de estampas de tempo (*time stamps*). Finalmente o dispositivo de leitura, que geralmente é um software, recolhe a informação do contador para informar ao sistema.

Sabendo como um relógio de computador funciona seguem algumas propriedades do tempo gerado por esses relógios:

- Deslocamento (*offset*) - é a diferença de tempo entre dois relógios.
- Dispersão (*dispersion*) - é o erro estimado nas leituras do relógio.
- Envelhecimento (*aging*) - quando a frequência do relógio varia com o tempo.
- Escorregamento (*drift*) - é a instabilidade da frequência do oscilador.

- Estabilidade (*stability*) - é a estimativa estatística da instabilidade do oscilador.
- Exatidão (*accuracy*) - é quanto o relógio está próximo da referência.
- Monotocidade (*monotonicity*) - o tempo deve sempre avançar entre uma leitura e outra.
- Precisão (*precision*) - o valor do menor incremento realizado pelo contador do relógio
- Sincronização (*synchronization*) - é o processo de ajustar dois relógios até a diferença entre eles ser zero.
- Sintonização (*syntonization*) - é o processo de ajustar dois osciladores para que oscilem na mesma frequência.
- Variação (*jitter*) - é o desvio ou erro nas leituras do relógio.

No caso do protocolo OWAMP o interesse maior será nas propriedades de exatidão e variação pois são estas que informam quanto um relógio está certo e quanto as medidas variam de um tempo para outro. Em suma busca-se um relógio com uma exatidão bastante alta e com variação pequena para que as medidas de atraso unidirecional sejam mais corretas possível.

2.5 Network Time Protocol

O *Network Time Protocol* (NTP) (RFC1305, 1992) é um protocolo para sincronização de relógios de computadores através da rede mundial de computadores que roda sobre UDP (RFC768, 1980). O fornecimento do horário é baseado em uma fonte confiável de tempo: usualmente um aparelho de *Global Positioning System* (GPS) ou um relógio atômico.

O funcionamento do NTP parece algo tão simples quanto requisitar de tempos em tempos o horário à um servidor e ajustar o relógio da máquina local, mas de na verdade vai muito além disso.

A partir de diversas amostras no decorrer do tempo são calculadas várias características do servidor, como deslocamento, dispersão e variação. Para isso acorrer é necessário modelar os parâmetros da rede que liga as máquinas. É preciso que o NTP saiba discernir servidores confiáveis e corretos daquelas que estão mentindo e dentre esses escolher aquele que forneça a melhor referência.

É necessário também entender o relógio local descobrindo seus principais parâmetros de funcionamento, como precisão, estabilidade e escorregamento para ajustá-lo de forma contínua e gradual, mesmo na ausência de fontes de referência confiáveis.

Além disso ele precisa garantir a monotonicidade do tempo e formar, em conjunto com outros servidores NTP, uma topologia simples, confiável, robusta e escalável para a sincronização do tempo.

Todas essas amostragens não são feitas um única vez, pelo contrário são efetuadas constantemente para que haja uma sintonia fina da precisão e para que eventuais erros sejam detectados. Portanto quanto mais tempo um servidor e um cliente ficarem conectados maior a probabilidade de eles estarem com os relógios ajustados.

2.5.1 A topologia do NTP

O NTP está estruturado em formato de árvore, ou seja, um servidor de hora pode ser, também, um cliente de algum outro servidor. Como mostrado na figura 2.2 os níveis desta árvore são chamados de estratos que são numerado de zero a dezesseis. O estrato zero é reservado às fontes de tempo, podendo não ser um computador (são dispositivos de informação de tempo). O estrato dezesseis indica que o protocolo está inoperante no computador. Logo, o número de um estrato indica o grau de profundidade que ele se encontra nessa árvore.

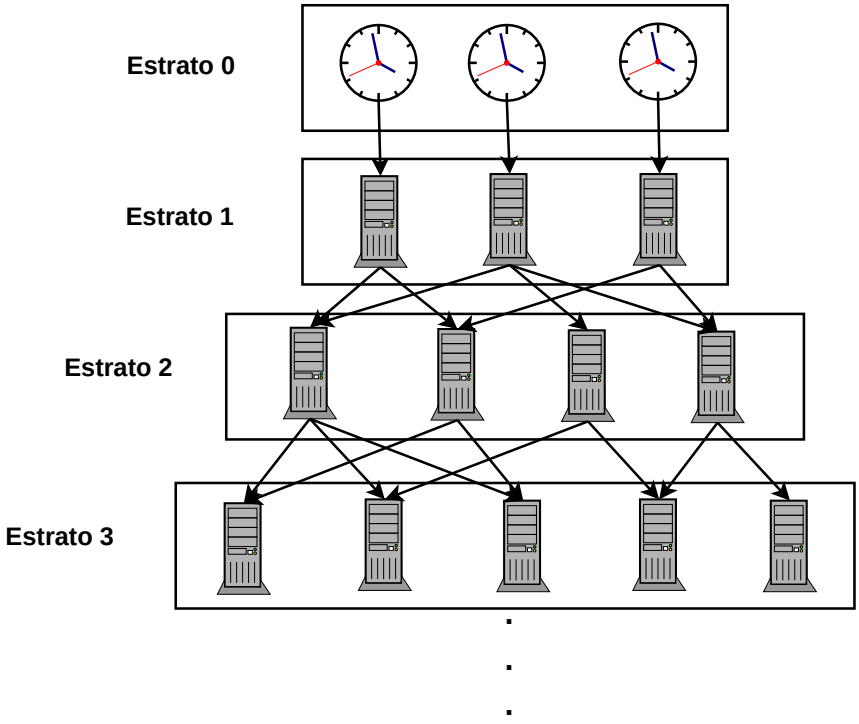


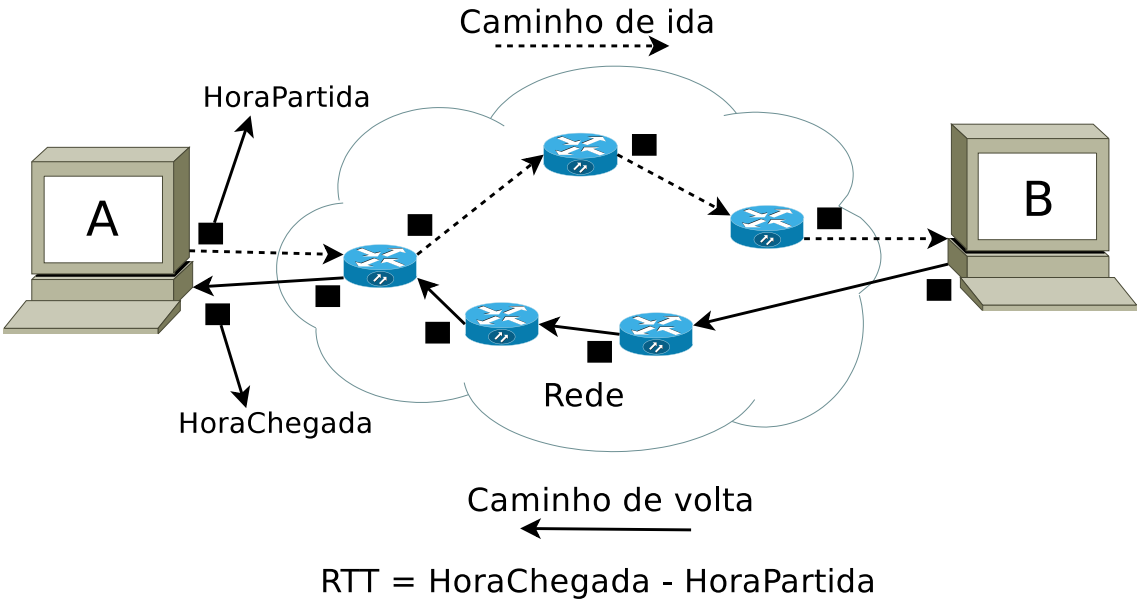
Figura 2.2: Topologia do NTP

2.6 Medidas *one-way* e *two-way*

Protocolos de medição baseados em *Round Trip Time* (RTT) são largamente utilizados na Internet, principalmente devido à facilidade de implementação e a interoperabilidade entre as máquinas. Estes fornecem medições que consideram a soma dos tempos de ida e volta do tráfego, sendo assim chamados de protocolos *two-way*.

Na figura 2.3 são mostrados dois computadores, A e B, interligados por uma rede. O computador A deseja efetuar um teste com a finalidade de medir o tempo que um pacote leva para ir até B e retornar. Um pacote é enviado de A para B contendo um campo com a hora de envio (“HoraPartida"na figura). O computador B recebe este pacote e envia de volta para A. Quando A recebe o pacote marca-o com o tempo de chegada (“HoraChegada"na figura). De posse dos horários de envio e recebimento, o computador A calcula o tempo que o pacote demorou para ir e voltar efetuando uma operação de diferença entre a hora de chegada e a hora de envio. Têm-se então os dados de uma medição *two-way*.

Protocolos *two-way* não possibilitam identificar a direção que está ocorrendo eventu-



.

Figura 2.3: Protocolos *two-way*

ais congestionamentos. Para resolver este problema foram propostos protocolos do tipo *one-way*, que são capazes de efetuar medições unidirecionais, ou seja, em apenas uma direção do tráfego. Um desses protocolos é o *One-Way Active Measurement Protocol* (OWAMP) que será alvo de estudo desse trabalho.

Um exemplo de programa *two-way* é o PING, que foi criado por Mike John Muuss em apenas uma noite no ano de 1983 visto que ele precisava resolver um problema na rede do laboratório em que trabalhava (MUUSS, 1983). Ele mesmo ficou impressionado com a popularidade que o programa ganhou. Ainda hoje o comando PING é uma ferramenta muito utilizada para execução de testes em redes e vem implementado por padrão em várias plataformas.

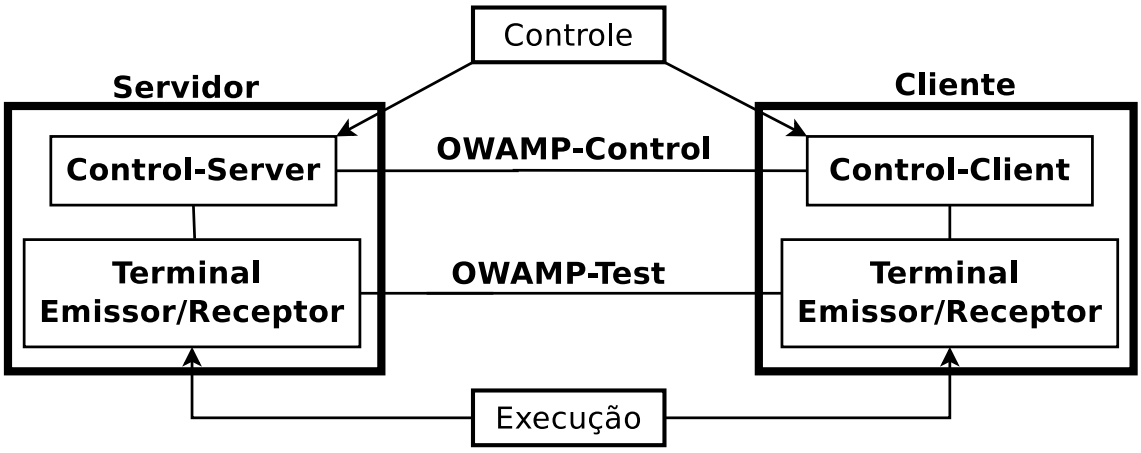
O PING é um programa que se vale de medidas *two-way*, ou seja, afere tempos que um determinado dado leva para ir de um ponto a outro e retornar. Sendo uma ferramenta largamente utilizada será neste trabalho explorado para efetuar comparação com o protocolo *one-way* proposto pelo o OWAMP.

2.7 One-way Active Measurement Protocol

O *One-way Active Measurement Protocol* (OWAMP) (RFC4656, 2006) é um protocolo interoperável de medições unidirecionais como atraso (RFC2679, 1999) e perda de pacotes (RFC2680, 1999) na Internet. O protocolo em questão faz parte de um conjunto de normas de medição em redes IP (RFC791, 1981) conhecido como *IP Performance Metrics* (IPPM) (RFC2330, 1998).

A especificação do OWAMP define que o protocolo seja dividido em sistema de controle e sistema de execução de testes. A definição do OWAMP apresenta dois protocolos inter-relacionados: *OWAMP-Control* e *OWAMP-Test* (detalhes na subseção 2.7.2 e 2.7.1). A figura 2.4 mostra uma visão desta topologia. Nela observa-se a existência de duas entidades, uma cliente e outra servidor, e a ligação entre elas que é efetuada por meio desses dois protocolos (*OWAMP-Control* e *OWAMP-Test*). Cada uma dessas entidades

está dividida em duas unidades distintas: controle e execução.



.

Figura 2.4: Topologia do protocolo OWAMP

Ambos os protocolos podem ser executados em três diferentes modos: sem-autenticação, autenticado e com criptografia. No caso de execução no modo autenticado ou com criptografia é necessário que as duas entidades que negociam a conexão possuam uma chave secreta compartilhada. A preocupação com segurança existe para evitar que os sistemas intermediários da rede alterem os pacotes de teste, o que tornaria as medições inconsistentes.

2.7.1 OWAMP-Control

Transportado via *Transmission Control Protocol* (TCP) (RFC793, 1981), o *OWAMP-Control* é o protocolo responsável pela negociação e controle da sessão de testes, bem como recuperação dos resultados obtidos. As principais funções executadas pelo *OWAMP-Control* são:

- Configuração da conexão;
- Criação de sessões de teste;
- Agendamento de horário para sessões de teste;
- Inicialização das sessões de teste;
- Encerramento das sessões de teste;
- Coleta de resultados.

2.7.2 OWAMP-Test

Sendo transportado pelo *User Datagram Protocol* (UDP) (RFC768, 1980) o *OWAMP-Test* é o protocolo que rege a troca dos pacotes entre os nós de medição. Para tal ele utiliza os parâmetros negociados durante a fase de configuração de conexão pelo protocolo *OWAMP-Control*.

A seguir será apresentado o comportamento do emissor e do receptor e um detalhamento dos pacotes usados para teste.

Emissor

Durante a configuração da sessão de testes pelo *OWAMP-Control* são pré-estabelecidos os momentos exatos que cada pacote de teste deve ser enviado. É como uma agenda de envio para todos os pacotes que serão enviados pelo *OWAMP-Test*. Esta agenda é passada para o emissor e receptor. Isto possibilita que eles efetuem suas operações independentemente, já que ambos sabem quando os pacotes serão enviados. Esses horários previamente acordados são independentes para cada sessão de teste, mesmo que controladas pela mesma sessão do *OWAMP-Control*.

Após uma sessão de teste ser configurada o emissor começa e enviar os pacotes, geralmente o tempo que sucede a configuração até o início dos envios é menor que um segundo. Os envios devem ocorrer o mais próximo possível do horário previamente acordado, a única exceção é se o emissor está atrasado em mais de *timeout* unidades de tempo. *Timeout* é um parâmetro, expresso em tempo, configurado pelo *OWAMP-Control* durante a configuração da sessão. Ou seja, o emissor não deve enviar um pacote que deveria ter sido mandado no passado se já passaram mais de *timeout* unidades de tempo da hora marcada. Embora não seja enviado, as informações deste pacote são armazenadas pelo emissor a fim de contabilizar os pacotes que foram perdidos na transmissão, já que o receptor vai considerar esse pacote como perdido na rede. Então, no momento da análise dos resultados é possível fazer a contabilização dos pacotes enviados, dos não enviados e dos perdidos.

Independentemente de eventuais atrasos, os pacotes devem ser marcados com um *timestamp* mais próximo possível do tempo real em que foram enviados. O emissor deve definir o *Time to Living* (TTL) (RFC791, 1981) (ou *Hop Limit* para IPv6 (RFC2460, 1998)) com o valor de 255 no pacote UDP. Na figura 2.5 é apresentado o esquema do formato de um pacote de teste no modo sem autenticação. Este pacote tem um tamanho mínimo de 14 bytes e é composto por quatro campos: número de sequência, *timestamp*, estimativa de erro e *packet padding*.

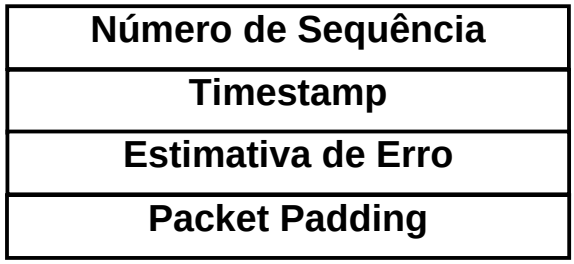


Figura 2.5: Pacote no modo sem autenticação

A figura 2.6 contém um diagrama de um pacote de teste para os modos com autenticação e com criptografia. O tamanho mínimo deste pacote é de 48 bytes e está dividido nos seguintes campos: número de sequência, 2 campos de MBZ, *timestamp*, estimativa de erro, HMAC e *packet padding*.

O formato do *timestamp*, mostrado na figura 2.7, está descrito na (RFC1305, 1992) e é da seguinte maneira: os 32 bits mais significativos representam um inteiro sem sinal com o tempo, em segundos, que se passaram desde as zero horas do dia primeiro de janeiro de 1900; os 32 bits menos significativos representam a fração de segundo que se passou desde o último segundo.

O campo estimativa de erro, mostrado na figura 2.8, é um parâmetro passado ao OWAMP pelo NTP (RFC1305, 1992) que indica o erro estimado, em segundos, do ho-

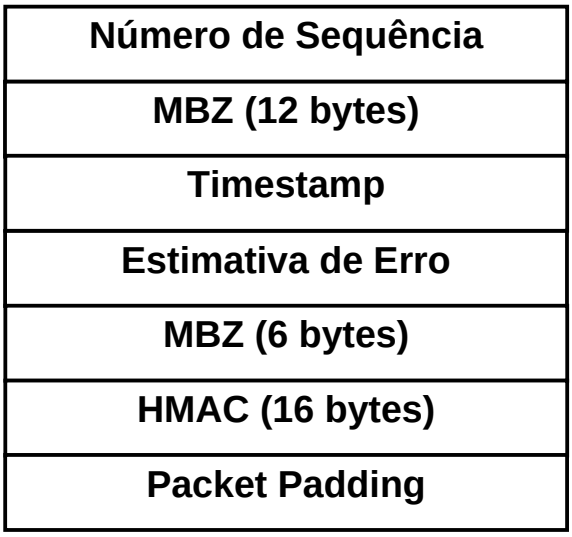


Figura 2.6: Pacote no modo com autenticação ou criptografia

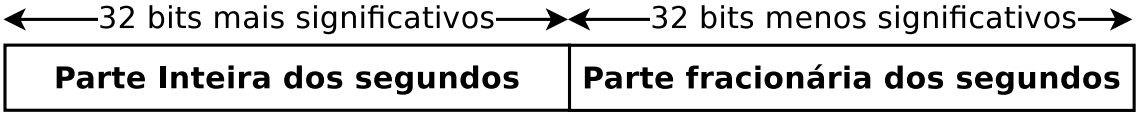


Figura 2.7: Detalhes do campo *timestamp*

rário do computador. É composto de 16 bits: O bit S indica, quando possui o valor 1, se o *timestamp* foi gerado por uma entidade sincronizada com alguma fonte de hora externa. O bit Z deve ser ajustado em zero pelo emissor e ignorado pelos demais. Os 6 bits de *scale* formam um inteiro sem sinal assim como os 8 bits do campo *multiplier*. Para obter o valor deste erro é necessário aplicar a seguinte fórmula sobre estes campos: $multiplier * 2^{-32} * 2^{scale}$. O campo *multiplier* não deve ser zero mas caso seja o pacote deve ser considerado corrompido e então descartado.

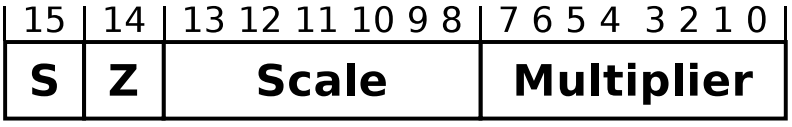


Figura 2.8: Detalhes do campo estimativa de erro

Número de sequência é zerado a cada nova sessão de testes e é incrementado após o envio de cada pacote subsequente. Os campos MBZ (*Must Be Zero*) são reservados para uso futuro e devem ser preenchidos com zeros pelo emissor e ignorado pelos demais.

O campo HMAC (*Keyed-Hashing for Message Authentication*) é descrito na (RFC2104, 1997) e é utilizado neste protocolo para os procedimentos de autenticação e criptografia. O campo de *packet padding* deve estar preenchido com todos os bits em zero. O número de bits deste campo deve ser aleatório, independe de qualquer outro número aleatório usado no protocolo, ou algum valor especificado pelo usuário.

Receptor

Este módulo é responsável pelo recebimento dos pacotes de teste. Dentre suas funções destacam-se:

- Armazenamento do *timestamp* dos pacotes, no momento de seu recebimento;
- Validação dos pacotes, caso seja utilizado o modo com autenticação ou criptografia;
- Gravação do número de sequência, hora de envio, hora de recebimento e TTL (ou *Hop Limit*).

No momento em que recebe um pacote, o receptor deve aplicar-lhe um *timestamp* para cálculos posteriores. Quando operando nos modos com autenticação e/ou criptografia é necessário que o receptor verifique a autenticidade e/ou remova a criptografia, respectivamente, de cada pacote. Caso algum pacote seja reprovado nessas operações ele deve ser descartado.

As informações dos pacotes válidos devem ser armazenadas para a contabilização dos resultados. É necessário guardar o número de sequência dos pacotes, para possibilitar o reordenamento, caso necessário; horário de envio e recebimento, para cálculo de atrasos; e o TTL (ou *Hop Limit*), para contabilização do número de nós pelos quais o pacote passou.

Existem algumas situações que os pacotes devem ser descartados, são elas:

- O *timestamp* de envio está expirado (*timeout* unidades no passado ou futuro);
- *Timestamp* de envio difere do horário marcado para envio por mais de *timeout* unidades de tempo;
- Falha na verificação de autenticidade e/ou remoção de criptografia.

2.7.3 Implementações disponíveis

Existem duas implementações disponíveis do OWAMP, são elas:

- OWAMP
- J-OWAMP

A primeira implementação (OWAMP) foi escrita em C para ambientes Unix pelo grupo *Internet2*, estando disponível para *download* no site (OWAMP, 2010). É a implementação de referência na área. Segundo seus desenvolvedores segue as instruções da (RFC4656, 2006) e está em constante aperfeiçoamento. Versões novas são liberadas frequentemente. Durante a realização deste trabalho a versão estável era a 3.1, já estando disponíveis algumas *releases candidates* da versão 3.2.

Como esta é a implementação de referência e seu nome coincide com o nome do procolo, qualquer menção posterior à ela será apenas escrita como "OWAMP", quando a referência for ao protocolo OWAMP, esta será especificada.

A segunda (J-OWAMP) foi implementada na linguagem JAVA pelo Instituto de Telecomunicações da Universidade de Aveiro em Portugal (J-OWAMP, 2010) . É uma alternativa à implementação da *Internet2*. Segundo os autores esta aplicação é totalmente compatível e interoperável com a aplicação OWAMP (Veiga, 2005).

3 ESPECIFICAÇÃO DOS TESTES

3.1 Descrição dos testes

Os testes executados tem o objetivo de verificar o funcionamento das duas implementações do protocolo OWAMP e em seguida comparar a implementação do grupo Internet 2 e o clássico programa PING, ou seja, comparar medições *one-way* e *two-way*.

Os testes estão divididos em quatro grupos independentes e cada um desses grupos não interfere nos resultados dos outros. Portanto não é necessário que eles sejam executas nesta ordem apresentada.

Os grupos são:

1. OWAMP IPv4
2. OWAMP IPv6
3. OWAMP *versus* J-OWAMP
4. OWAMP *versus* PING

Os grupos OWAMP IPv4 e OWAMP IPv6 correspondem ao testes de funcionalidade e precisão do programa OWAMP (implementação do grupo Internet2) em seu diferentes modos de operação (com autenticação, com criptografia e aberto). O grupo OWAMP *versus* J-OWAMP corresponde aos testes comparativos entre as duas implementações do protocolo OWAMP. A categoria OWAMP *versus* PING apresenta o comparativo do programa *one-way* OWAMP (implementação do grupo Internet2) e do programa *two-way* PING.

Para todos os testes foram enviados um total de um mil pacotes entre o emissor e o receptor. Todos os testes foram executados em separado, ou seja, cada um rodando sozinho no computador. Por exemplo: primeiro foi executado o teste do OWAMP IPv4 no modo aberto, depois deste finalizado é que foi rodado o teste do OWAMP IPv4 no modo com autenticação. Além disso o computador além de rodar o sistema operacional e os programas necessários para o seu funcionamento só estava rodando os programas relativos ao teste (OWAMP e NTP).

Estas medidas foram adotadas para que houvesse o mínimo de interferência possível de fatores externos na execução dos testes. Como o funcionamento do protocolo OWAMP e do próprio NTP é muito sensível ao tempo estas medidas são realmente necessárias. Não se pode executar esses testes com máquinas virtuais, por exemplo, porque isso introduziria muitos implicantes negativos no correto funcionamento dos programas em questão.

Para o correto funcionamento do NTP é necessário que o programa receba um sinal de relógio da máquina em que está rodando e dependendo da aplicação utilizada para a

virtualização, este sinal é recebido do sistema operacional hospedeiro da máquina virtual e não diretamente da hardware, o que introduziria um atraso e variação dessas leituras muito grande. Sem o NTP funcionando perfeitamente esses erros refletiriam diretamente nas medições do programa OWAMP.

Em todos os grupos foi configurada uma política de gerenciamento de filas do tipo *netem* (*Network Emulator*) na saída do roteador. Isso foi feito para criar um ambiente imitando um situação real, pois como o cenário era exclusivo para estes testes não havia nenhum outro tipo de tráfego nesses enlaces. Esta política funciona, com os parâmetros utilizados, basicamente adicionando um atraso em todos os pacotes que passam por ela, ou seja, cada pacote aguarda na fila um tempo determinado, conforme configurado, antes de ser enviado.

Essa política é bastante interessante pelo fato de ser possível escolher a ordem de grandeza a ser trabalhada. Sem esta política, com o cenário exclusivo para os testes, os atrasos medidos eram inferiores a um milissegundo e dessa maneira dificultavam a visualização do correto funcionamento dos programas analisados.

3.2 Equipamento utilizado

Os equipamentos utilizados foram:

- Computador A: equipado com processador AMD Athlon 64 X2, 2GB de memória RAM, 2 interfaces de rede e rodando a ferramenta pS-performance (pS-performance, 2010);
- Computador B: equipado com processador AMD Athlon 64 X2, 1Gb de memória RAM, 5 interfaces de rede e rodando Linux Ubuntu 10.10 kernel 2.6.34;
- Computador C: equipado com processor Intel Core i5, 4GB de memória RAM, 2 interfaces de rede e rodando a ferramenta pS-performance (pS-performance, 2010);
- 5 cabos de rede padrão CAT 5E.

3.3 Cenário de testes

Os computadores A e C funcionaram como emissor e receptor dos testes, respectivamente. O computador B funcionou como o roteador e como o servidor de hora NTP, que estava sincronizado por meio da Internet com os computadores do projeto NTP do Brasil que são servidores de estrato um, logo o servidor de hora usado nos experimentos era uma servidor de estrato 2.

O computador B estava ligados aos computadores A e C por meio de quatro cabos de rede, dois para o emissor (A) e dois para o receptor (B). Cada conexão pertencia à uma sub-rede diferente sendo elas:

- 10.2.2.0/24 - 2002::/32: Conexão entre o emissor e roteador para envio de pacotes de teste;
- 10.3.3.0/24 - 2001::/32: Conexão entre o roteador e receptor para recebimento dos pacotes de teste;
- 10.4.4.0/24: Conexão para sincronismo de relógio entre o roteador e emissor;

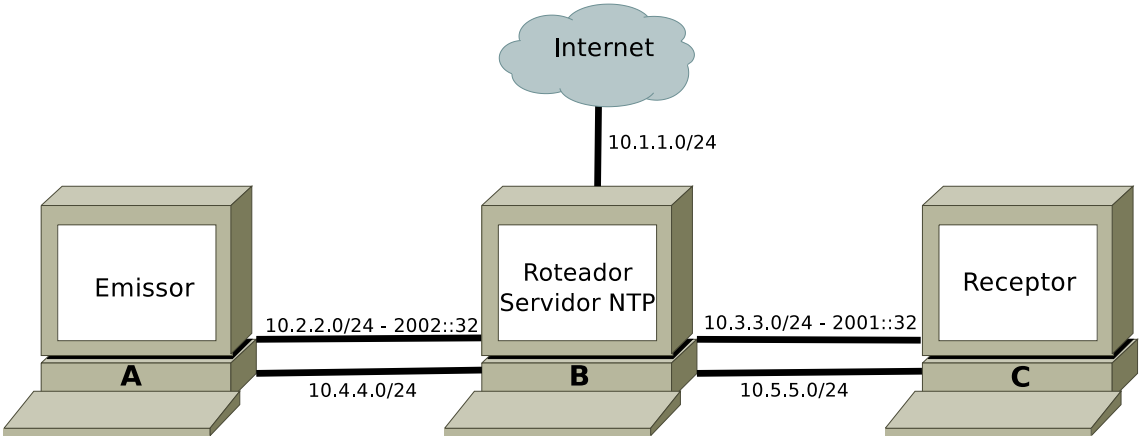


Figura 3.1: Cenário de teste

- 10.5.5.0/24: Conexão para sincronismo de relógio entre roteador e receptor.

Além dessas sub-redes havia mais uma para conexão do roteador com a Internet para o sincronismo de hora, a qual era a rede 10.1.1.0/24. Todas essas conexão estão mostradas na figura 3.1.

No capítulo a seguir serão apresentados os resultados das baterias de testes dos grupos um ao quatro.

3.4 Configuração das máquinas e ferramentas utilizadas

Utilizando a distribuição *ps-Performance Toolkit* a configuração do OWAMP e do NTP ficam bastante facilitadas, pois esses já vêm instalados e com uma configuração padrão.

Para o funcionamento do OWAMP é necessário o NTP estar configurado, rodando e com o horário sincronizado. Para configurar o NTP basta editar o arquivo */etc/ntp.conf*, adicionar os servidores de hora a serem utilizados e após reinicializar o *daemon ntpd*. No caso deste trabalho, o IP do computador que funcionava como roteador foi o servidor NTP descrito no arquivo de configuração.

Após está configuração o OWAMP está pronto para funcionar. Apenas é necessário configurar os usuários e senha para operação em modo com autenticação e com criptografia. Para fazer isso é necessário utilizar o comando *aespasswd* descrevendo o usuário e o arquivo onde esta configuração ficará armazenada. Segundo o manual do OWAMP este arquivo deve estar no diretório de configuração do OWAMP, juntamente com o arquivo *owamp.conf*, e ser chamado de *owamp.keys*. Para que uma bateria de testes seja executada entre dois nodos da rede é necessário que ambos possuam este arquivo com o mesmo nome de usuário e senha.

A ferramenta básica para testes do OWAMP é o *owping*, uma ferramenta muito semelhante ao PING. Com este comando é possível configurar a sessão de testes com parâmetros como: IP destino, número de pacotes de teste, modo de operação, etc. Para efetuar um teste simples de funcionamento do OWAMP basta executar o comando *owping localhost*, como este comando será executado um teste com cem pacotes diretamente na interface do próprio computador.

Funcionamento do OWAMP

Algumas observações precisam ser feitas sobre o OWAMP: embora na maioria da vezes o funcionamento foi estável, muitas vezes ocorreram problemas na configuração da sessão de testes, por motivos desconhecidos o comando *owping* simplesmente não funcionava em determinados momentos. Geralmente neste caso a simples re-execução do comando resolvia o problema, mas algumas vezes foi necessário a reinicialização das máquinas para que ele voltasse ao funcionamento normal.

Embora este não tenha sido um grande empecilho, por muitas vezes os testes ficaram atrasados, pois o erro não aparecia rapidamente na tela e sim demorava algum tempo considerável para ser exibido. Em alguns momentos o problema só era solucionado com a reinicialização de todas as máquinas do cenário de testes, o que demandava um tempo razoável para as configurações de todas as interfaces de rede e verificações de funcionamento.

4 EXECUÇÃO E RESULTADOS DOS TESTES

4.1 Testes IPv4

Nesta seção encontram-se os resultados dos testes do programa OWAMP utilizando pacotes sobre IPv4. Todos os experimentos foram efetuados com o envio de 1000 pacotes de teste. No roteador estava configurada uma política de gerenciamento de despacho de pacotes, do tipo NETEM, na interface de saída para acrescentar atraso na rede, o valor de atraso escolhido foi de 100 ms. Todas as medidas levaram e consideração um índice de confiança de 95%.

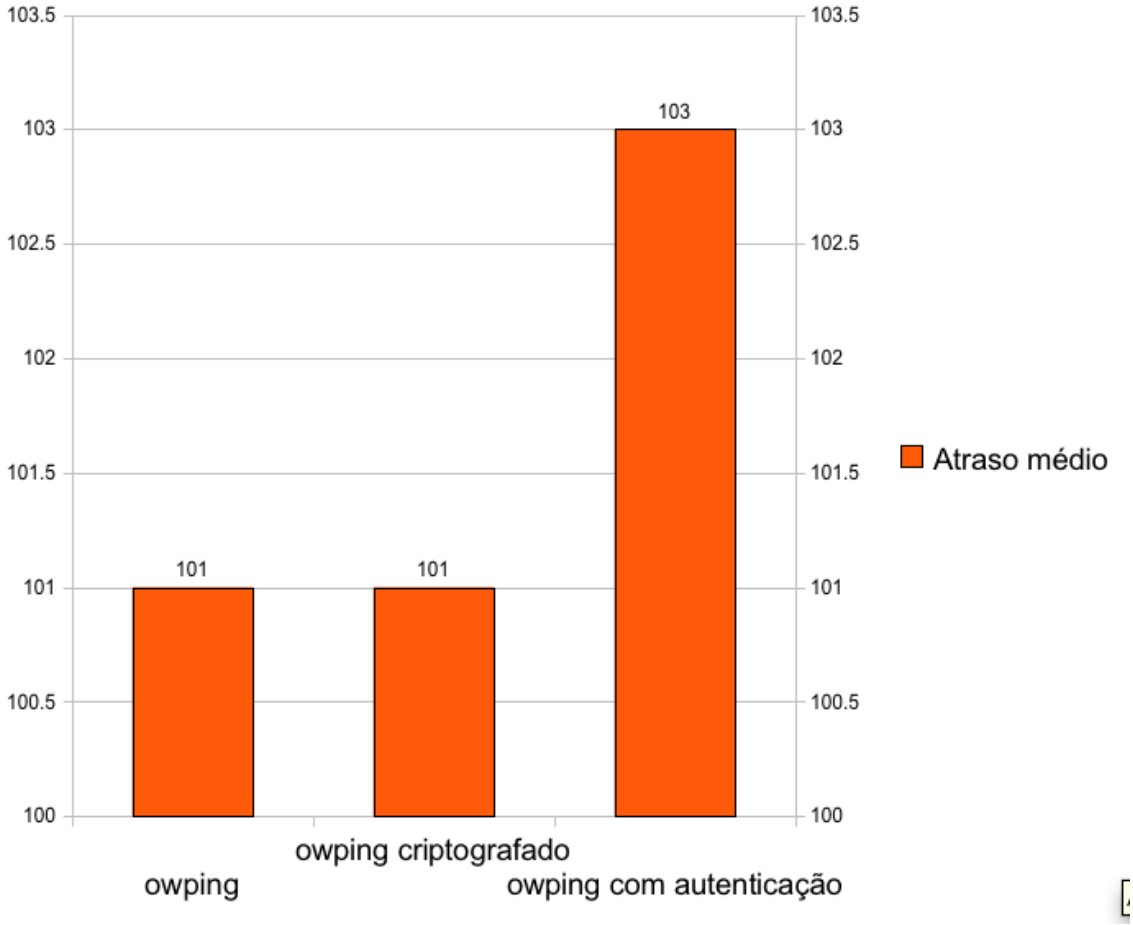


Figura 4.1: Atrasos IPv4 - Descrição do teste *versus* Tempo(ms)

O gráfico da figura 4.1 mostra as medições efetuadas nos diferentes modos de ope-

ração do OWAMP. É possível notar que os atrasos médios aferidos tanto para o modo criptografado e sem autenticação foi de 101 ms enquanto que para o modo com autenticação o valor foi de 103 ms. Considerando que o atraso incluído no caminho apenas pela política de fila foi de 100 ms é possível inferir que os tempos extras (1 ms ou 3 ms) foram gastos na transmissão e decodificação dos pacotes.

Teste	Atraso médio (ms)	Mínimo - Máximo (ms)	Intervalo de confiança - 95%
OWPING com autenticação IPv4	103	101 - 107	93.98 - 112.02
OWPING com criptografia IPv4	101	96.7 - 104	94.53 - 107.47
OWPING IPv4	101	96 - 102	100.61 - 101.39

Tabela 4.1: Resultados dos testes IPv4

A tabela 4.1 apresenta os valores de atraso médio e intervalo de confiança para um índice de confiança de 95% na medição dos 1000 pacotes de teste. Com os valores apresentados a conclusão é de que os modos com autenticação ou criptografia são menos precisos que o aberto. A tabela mostra os intervalos de confiança bem maiores para esses dois casos, já para a execução em modo aberto a precisão é consideravelmente boa apresentando uma variação, com um índice de confiança de 95%, menor que 0,4% em torno do valor médio.

4.2 Testes IPv6

Nesta seção encontram-se os resultados dos testes do programa OWAMP utilizando pacotes sobre IPv6. Todos os experimentos foram efetuados com o envio de 1000 pacotes de teste. No roteador estava configurada uma política de gerenciamento de despacho de pacotes, do tipo NETEM, na interface de saída para acrescentar atraso na rede, o valor de atraso escolhido foi de 100 ms. Todas as medidas levaram e consideração um índice de confiança de 95%.

O gráfico da figura 4.2 mostra as medições efetuadas nos diferentes modos de operação do OWAMP. É possível notar que os atrasos médios aferidos tanto para o modo criptografado e sem autenticação foi de 103 ms enquanto que para o modo com autenticação o valor foi de 99 ms. Considerando que o atraso incluído no caminho apenas pela política de fila foi de 100 ms é possível inferir que o tempo extra de 3 ms foi gasto na transmissão e decodificação dos pacotes, já para o caso do valor 99 ms este é justificado por uma pequena interferência no sincronismo de relógio entre as máquinas.

Teste	Atraso médio (ms)	Mínimo - Máximo (ms)	Intervalo de confiança - 95%
OWPING com autenticação IPv6	99	99.8 - 100	95.88 - 102.12
OWPING com criptografia IPv6	103	98.8 - 103	99.98 - 106.02
OWPING IPv6	103	103-104	99.73 - 106.27

Tabela 4.2: Resultados dos testes IPv6

A tabela 4.2 apresenta os valores de atraso médio e intervalo de confiança para um índice de confiança de 95% na medição dos 1000 pacotes de teste. Com os valores apre-

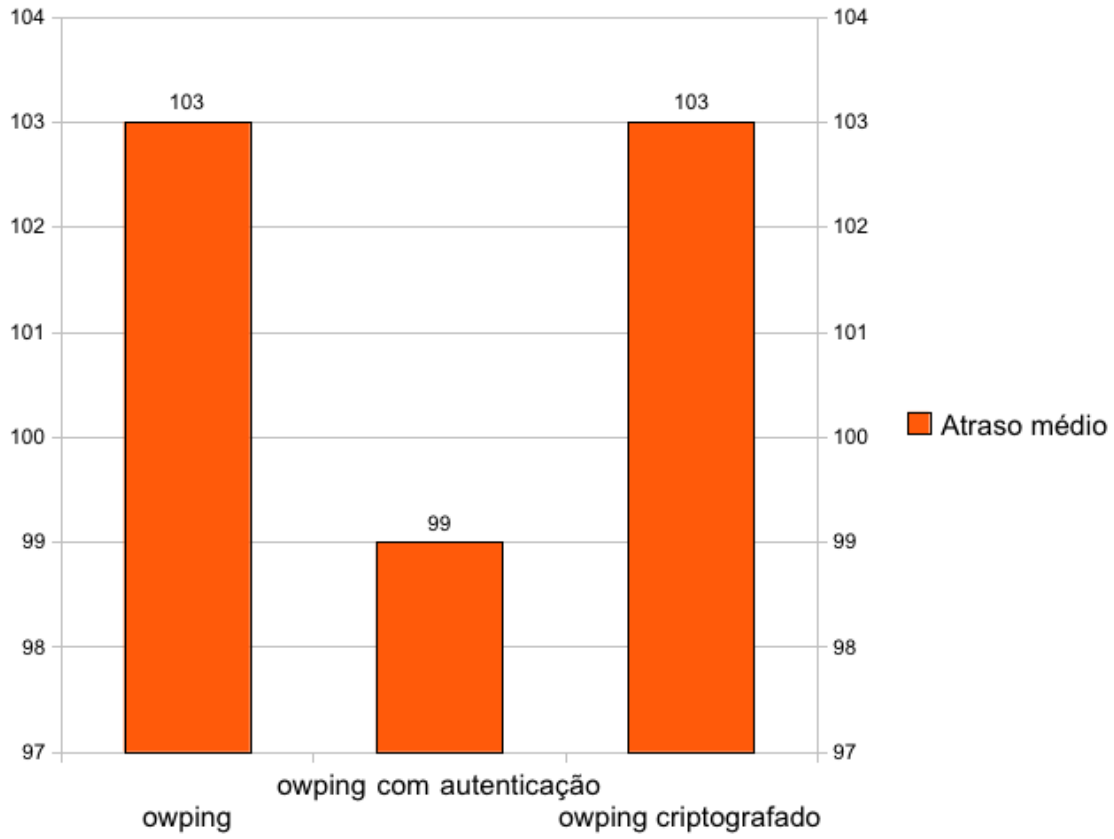


Figura 4.2: Atrasos IPv6 - Descrição do teste *versus* Tempo(ms)

sentados a conclusão é de que as medidas tiveram pouca variação. Para o modo com autenticação o intervalo de confiança ficou entre 95.88 ms e 102.12 ms, para o modo com criptografia ficou entre 99.98 ms e 106.02 ms e para o modo aberto entre 99.73 ms e 106.27 ms. Esse valores apresentam uma variação em torno de média de aproximadamente 3% para os três casos.

4.3 OWAMP versus J-OWAMP

Nesta seção são encontradas as medidas comparativas entre as duas implementações do protocolo OWAMP existentes, a implementação do grupo da Internet 2 e a implementação da Universidade de Aveiro. Novamente todos os experimentos foram executados com a emissão de 1000 pacotes de teste e no roteador estava configurada uma política de gerenciamento de despacho de pacotes, do tipo NETEM, na interface de saída para acrescentar atraso na rede, o valor de atraso escolhido foi de 100 ms. Todas as medidas levaram e consideração um índice de confiança de 95%.

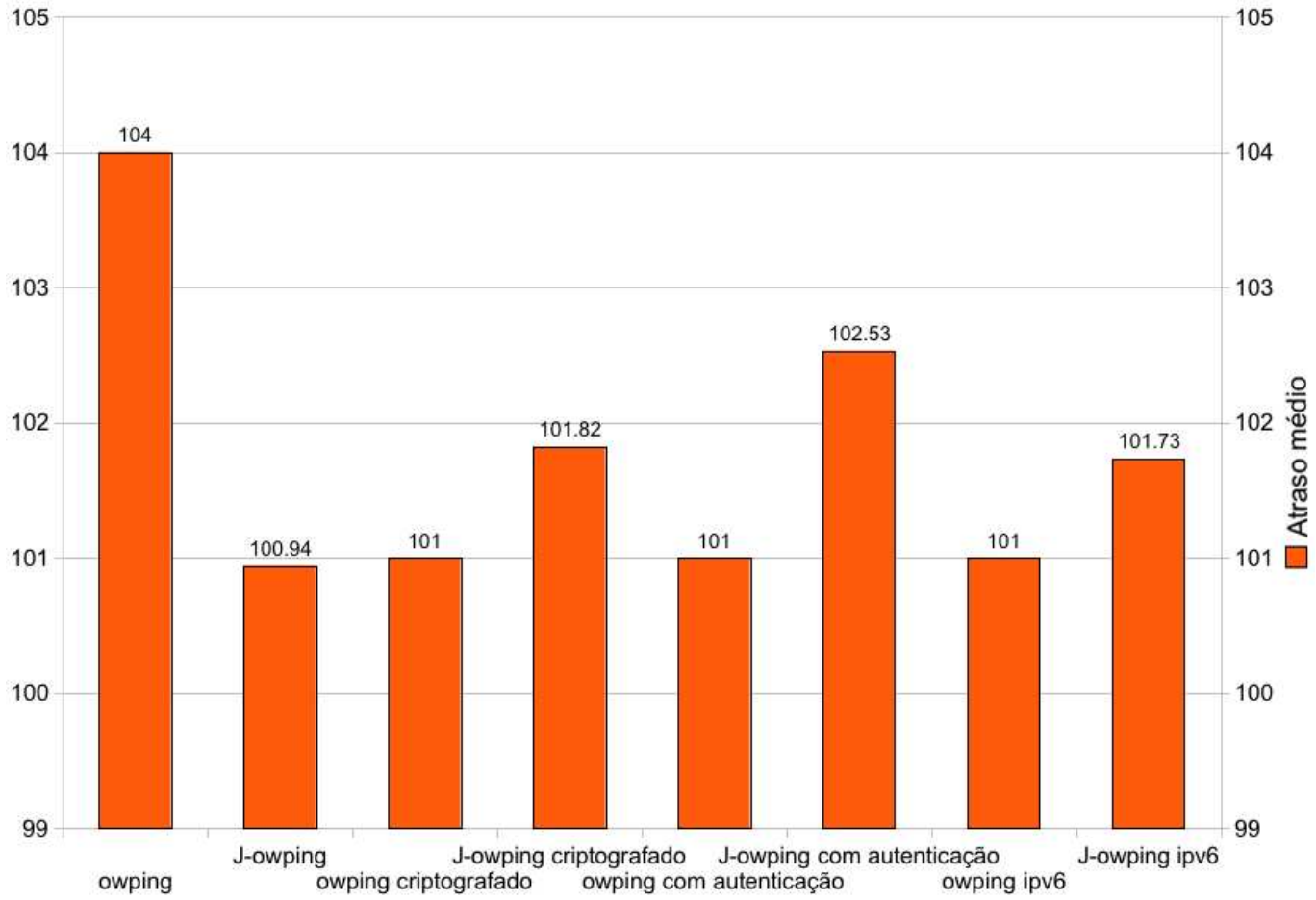


Figura 4.3: Atrasos J-OWAMP *versus* OWAMP - Descrição do teste *versus* Tempo(ms)

No gráficos da figura 4.3 são encontrados os valores para diversos modos de funcionamento das implementações sobre IPv4 e o teste sobre IPv6 apenas da versão sem autenticação. Observando os valores é possível notar que as duas implementações fornecem resultados bastante próximo. O atraso médio para um mesmo modo de operação e sobre o mesmo protocolo IP diferem em no mínimo 0.72% e no máximo em 3%.

Teste	Atraso médio (ms)	Mínimo - Máximo (ms)	Intervalo de confiança - 95%
OWPING IPv6 (J-OWAMP)	101	101-101	100.61 - 101.39
J-OWPING IPv6	101.73	101.37-105.59	101.12 - 102.34
OWPING IPv4 (J-OWAMP)	104	104-104	103.61 - 104.39
J-OWPING IPv4	100.94	100.33-104.04	99.9 - 101.98
OWPING IPv4 com autenticação (J-OWAMP)	101	100.07-104.02	100.61 - 101.39
J-OWPING IPv4 com autenticação	102.53	101-101	101.55 - 103.51
OWPING IPv4 com criptografia (J-OWAMP)	101	101.22-102.68	96.49 - 105.51
J-OWPING IPv4 com criptografia	101.82	99.8-103	100.84 - 102.8

Tabela 4.3: Resultados dos testes J-OWAMP *versus* OWAMP

Analisando tabela 4.3 é possível notar que os testes, exceto o modo com criptografia

do OWAMP, fornecem a possibilidade de inferência que para um índice de confiança de 95% os valores medidos oscilam em torno do valor médio em mais ou menos 0.4 a 1 ms. No caso diferente, *owping* com criptografia sobre IPv4, o intervalo de confiança fica em 4.51 ms para mais ou para menos em relação ao valor médio.

4.4 OWAMP versus PING

Nestão seção é mostrado um comparativo entre o OWAMP e o tradicional programa PING. Em cada experimento foram enviados um total de 1000 pacotes de testes. Neste teste é adicionado atraso no roteador nas interfaces de saída em ambos os sentidos, na ida um valor de 100ms e na volta um valor de 50ms. Todas as medidas levaram e consideração um índice de confiança de 95%.

O gráfico da figura 4.4 apresenta as medidas de atrasos do teste executado. Pode-se notar os valores de ida e de volta do programa OWAMP, esses valores somados e os valores do programa PING. É interessante verificar que o somatório dos valores médios aferidos pelo OWAMP e o valor médio contabilizado pelo PING difere por menos de 0.2%. Isto mostra que o OWAMP é um programa com exatidão bastante alta e o fato das componentes da soma serem diferentes (50 ms e 100 ms) aponta para uma das mais importantes utilidades das medidas *one-way* que é verificar em qual caminho exato da rede está acontecendo o congestionamento mais significativo, ou seja, apontar o gargalo da comunicação.

Teste	Atraso médio (ms)	Mínimo - Máximo (ms)	Intervalo de confiança - 95%
OWPING IPv4 ida (PING)	101	96-102	100.61 - 101.39
OWPING IPv4 volta (PING)	48.9	48.6-54.2	48.37 - 49.43
OWPING IPv4 soma (PING)	149.9	144.6-156.2	149.44 - 150.36
PING	150.48	150.18-150.58	150.28 - 150.68

Tabela 4.4: Resultados dos testes PING

A tabela 4.4, que apresenta os valores do atraso médio e do intervalo de confiança para as medidas, indica que o OWAMP tem uma variação nas medidas um pouco maior que o PING. Mas como visto anteriormente isso não invalida a precisão do OWAMP, apenas informa que, dependendo da situação, para medidas mais precisas é necessário a utilização de maior quantidade de pacotes de teste. Embora essa variação seja maior, a diferença de valores nas medições entre o PING e o OWAMP foi inferior à 1%.

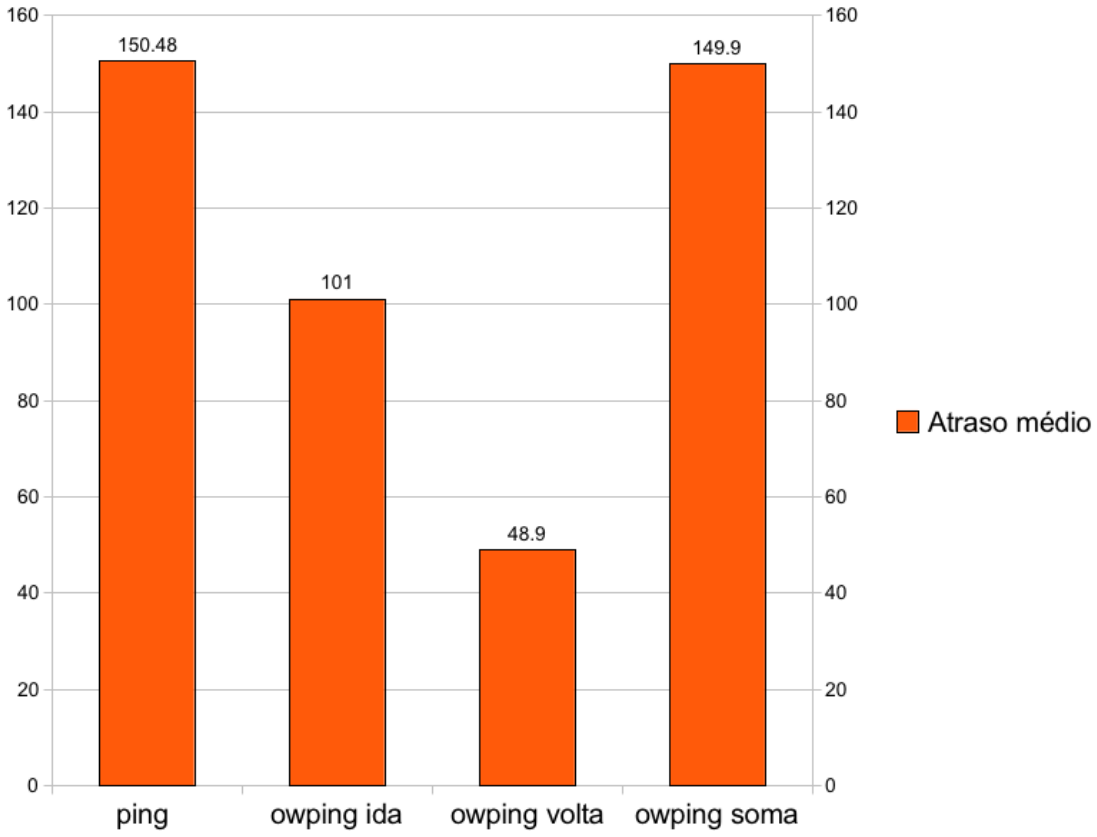


Figura 4.4: Atrasos PING *versus* OWAMP - Descrição do teste *versus* Tempo(ms)

5 CONCLUSÃO

A utilização de um protocolo capaz de medir tempos em uma direção, para identificar pontos críticos de congestionamento num enlace de dados, é uma abordagem bem interessante. A utilização de todos os programas pretendidos ocorreu com sucesso, entre eles o OWAMP, NTP, J-OWAMP e PING, inclusive as variações nos funcionamentos de cada um. Com os testes foi possível identificar características bem interessantes de cada um.

O protocolo PING devido a sua simplicidade, desempenho e disseminação apresentou resultados bastante consistentes e foi inclusive uma boa ferramenta para durante o trabalho realizar a conferência de alguns resultados. Na verdade a família ao qual este pertence, protocolos *two-way*, é muito mais fácil de ser implementada devido à não necessidade da sincronização de relógios entre as partes envolvidas.

A implementação OWAMP também apresentou resultados satisfatórios, porém é um tipo de programa bem mais difícil de ser configurado e apresenta alguns comportamentos estranhos (para de funcionar, não consegue negociar as sessões de teste, etc) sendo necessário ou a repetição do comando ou a reinicialização da máquina. Devido ao fato de ser necessário configurar criteriosamente o OWAMP é difícil que ele se torne popular, exceto em situações de redes de empresas de grande porte, empresas de telecomunicações por exemplo, onde o controle de toda rede é centralizado e assim fácil de instalá-lo nos nós desejados.

O J-OWAMP é uma ferramenta bem interessante pelo fato de poder rodar em qualquer sistema operacional com a plataforma JAVA instalada. Além disso possui por padrão uma interface gráfica que o torna uma excelente ferramenta para trabalhar como cliente nas sessões de testes, facilitando a execução rápida de testes pontuais. Analisando desempenho ele é bem próximo ao OWAMP, apenas apresentou comportamento estranho, impossibilidade de negociação da sessão de testes, quando executado um teste num enlace extremamente saturado de dados.

A descoberta mais relevante é que, quando comparado o OWAMP com o PING, o OWAMP apresenta uma precisão suficiente para identificar a direção em que os congestionamentos estão ocorrendo, ou em qual há um maior congestionamento de tráfego. Com base nisso o OWAMP pode ser utilizado como uma ferramenta auxiliar para avaliações de enlaces, ou seja, quando usado em conjunto com o PING fornece dados que detalham um pouco mais a situação de determinado ramo da rede.

Uma sugestão para trabalhos futuros é a criação de uma interface gráfica ou textual para o OWAMP onde fosse possível a configuração de sessões de teste mais elaboradas. Com isso seria possível monitorar determinado enlace durante o dia todo e traçar gráficos de comportamento da rede de acordo com o horário do dia, ou seja, montar um perfil de atrasos de determinado enlace.

REFERÊNCIAS

Carissimi, Alexandre da S., J. Rochol, L. Granville. . **Redes de Computadores**. Bookman, 2009

H. Veiga, J. Oliveira, R. Valadas, P. Salvador, A. Nogueira, and J. Silva. **J-OWAMP: Java Implementation of OWAMP**. http://www.av.it.pt/jowamp, acessado em junho de 2010.

Higginbottom, Gary N.. . **Performance Evaluation of Comunication Networks**. Artech House Inc., 1998

Muuss, Mike John. **The history of the PING program**. http://ftp.arl.army.mil/ mike/ping.html, acessado em novembro de 2010.

Internet 2. **One-Way Ping (OWAMP)**. http://www.internet2.edu/performance/owamp, acessado em junho de 2010.

Internet 2. **pS-performance toolkit**. http://www.internet2.edu/performance/toolkit/, acessado em novembro de 2010.

J. Postel. **RFC 768 - User Datagram protocol**. 1980

J. Postel. **RFC 791 - Internet Protocol**. 1981

J. Postel. **RFC 793 - Transmission Control Protocol**. 1981

Dave L. Mills. **RFC 1305 - Network Time Protocol (Version 3) - Specification, Implementation and Analysis**. 1992

H. Krawczyk, M. Bellare, and R. Canetti. **RFC 2104 - HMAC: Keyed-Hashing for Message Authentication**. 1997

V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. **RFC 2330 - Framework for IP Performance Metrics**. 1998

S. Deering, and R. Hinden. **RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification**. 1998

G. Almes, S. Kalidini, and M. Zekauskas. **RFC 2679 - A One-way Delay Metric for IPPM**. 1999

G. Almes, S. Kalidini, and M. Zekauskas. **RFC 2680 - A One-way Packet Loss Metric for IPPM**. 1999

S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, and M. Zekauskas. **RFC 4656 - A One-Way Active Measurement Protocol (OWAMP)**. 2006

H. Veiga, T. Pinho, J. Oliveira, R. Valadas, P. Salvador, and A. Nogueira. **Active traffic monitoring for heterogeneous environments**. 4th International Conference on Networking, ICN'05, 2005 – Reunion Island.