

Uma Biblioteca de Classes para Medição de Serviços Diferenciados

Alexandre Briani Kielling
briani@inf.ufsc.br

Elvis Melo Vieira
elvis@npd.ufsc.br

Carlos Becker Westphall
westphal@lrg.ufsc.br

RMAV-FLN
Núcleo de Processamento de Dados/Laboratório de Gerência e Redes - INF
Campus Universitário, Trindade
Universidade Federal de Santa Catarina
Florianópolis, SC – 88037-010

Engenharia de Rede: Medições, Qualidade de Serviço

Resumo

Este documento apresenta uma biblioteca de classes para medição de latência, variação de latência e perda de pacotes. Inicialmente ela está sendo usada no projeto de um Sistema de Gerência de Qualidade de Serviço chamado Mediator, no âmbito do projeto de “Qualidade de Serviço” da RMVA-FLN. Entretanto, está disponível para ser utilizada em outras aplicações de gerência ou aplicações distribuídas que necessitam que QoS.

1 Introdução

Existe uma série de requisitos de QoS (*Quality of Service*) que deverão ser cumpridos para a implementação da Internet2. Um destes requerimentos de QoS é a possibilidade de medição dos serviços disponíveis na rede. A existência de uma boa infraestrutura de medição é sempre importante, principalmente durante o período inicial de implementação de novos serviços. Através de ferramentas de medição e métricas de QoS bem definidas pode-se monitorar a performance garantida nos contratos de alocação de recursos da rede.

Existe uma grande possibilidade de que os Serviços Diferenciados sejam implementados na Internet2. Nesta arquitetura de QoS existem entidades responsáveis pela alocação dos recursos da rede. Esta entidade é chamada de *Bandwidth Broker* e deve existir uma em cada domínio de Serviços Diferenciados. As principais responsabilidades dos *Bandwidth Brokers* são: controle de admissão de acordo com políticas pré-definidas e configuração dos roteadores.

É provável que no futuro os *Bandwidth Brokers* tenham recursos de monitoração da QoS dos serviços disponibilizados. Através de métricas como latência, variação de latência e perda de pacotes, é possível visualizar mais claramente como os dados estão sendo transmitidos pela rede. Existem diversas ferramentas de geração de tráfego e monitoração disponíveis atualmente, porém não são muito flexíveis para a integração com outras aplicações. Uma biblioteca de classes (em C++, até o presente momento) de medição seria uma alternativa melhor para os desenvolvedores de *Bandwidth Brokers* e

ferramentas de medição do tipo cliente/servidor. Desta forma, o desenvolvimento destas ferramentas pode se tornar mais rápido. Além disso, as aplicações que recebem os serviços podem utilizar a mesma biblioteca de classes para verificar o desempenho da rede e, desta forma, certificar-se que os acordos de nível de serviços estão sendo mantidos.

2 A Biblioteca de Classes

A biblioteca de classes apresentada neste documento contém quatro classes, inicialmente disponibilizadas em C++. Duas delas são usadas para a geração de números randômicos de acordo com uma distribuição de Poisson. As outras duas são usadas para medições usando UDP e TCP. O resultado das medições é representado por um *array* contendo os valores de latência dos pacotes de teste. Os cálculos de tempo são realizados com precisão de microsegundos e os pacotes perdidos são representados pelo valor zero. A seguir cada classe é apresentada em mais detalhes.

2.1 Random

Esta classe tem a simples função de gerar valores randômicos requisitados através do método `nextDouble()`.

2.2 RandomPoissonDistribution

Esta classe gera valores de acordo com uma distribuição de Poisson. O valor médio da distribuição é definido no construtor da classe. Para gerar valores da distribuição de Poisson usa-se o método `nextLong()`, o qual devolve um valor no formato *long*.

2.3 UDPProbe

Esta classe contém os métodos de medição onde o tráfego é composto por pacotes UDP. Os métodos usados para medição são chamados `dispatch` e `probe`. O método `probe` é executado no *host* fonte e o método `dispatch` é executado no *host* destino. A troca de informações de controle entre os dois sistemas é realizada através de sockets TCP.

O método `probe` recebe os seguintes parâmetros : o endereço do *host* destino, as portas fonte e destino do tráfego de teste, o número de pacotes que serão transmitidos e o tamanho destes pacotes. Através destas informações é realizado o envio dos pacotes de teste para o *host* destino e, no final da medição, recebido um *array* de valores contendo a latência de cada pacote. Cada pacote tem um identificador para controle de sequência da transmissão.

O método `dispatch` implementa a funcionalidade requerida no sistema servidor e seu único parâmetro é o número da porta na qual receberá os pacotes de teste enviados pelo sistema cliente. Sua função é receber cada pacote, ler o tempo de envio do pacote e calcular a latência do mesmo. No final da medição, os valores de latência calculados são enviados de volta ao método `probe`.

2.4 TCPProbe

Esta classe tem a mesma funcionalidade da classe UDPProbe, porém realiza o envio de pacotes de teste do tipo TCP. Neste caso, a mesma conexão TCP para troca de informações de controle é usada para as transmissões dos pacotes de teste. Esta abordagem foi escolhida por motivo de simplicidade.

3 Procedimentos de Medição

O procedimento de medição implementado na biblioteca de classes foi baseado em documentos desenvolvidos pelo grupo de trabalho *IP Performance Metrics* (IPPM) do IETF [1], o qual define métricas para medição da performance de redes TCP/IP.

3.1 Método Poisson

Uma forma comum de medir a latência é coletar amostras regularmente e em tempos fixos. Este método é atrativo pela simplicidade, mas pode apresentar alguns problemas. Se a latência da rede apresentar comportamentos periódicos coincidentes com os momentos de amostragem, existe a possibilidade das amostras observarem somente parte do comportamento. Além disso, as medições periódicas e repetidas podem gerar perturbações na rede.

Um método mais efetivo, o qual é usado neste trabalho, é o baseado em amostras randômicas, onde os intervalos têm uma distribuição de Poisson. Normalmente, este método afeta menos a rede e gera uma estimativa mais imparcial da métrica em questão. Devido a estas características tão importantes, o método Poisson é o método preferido para a realização de medições na Internet. A amostragem Poisson é feita através da geração de intervalos distribuídos exponencialmente e realização de uma medição após cada intervalo decorrido. Cada amostragem é distribuída uniformemente no tempo total de medição.

3.2 Medição de Latência One-way

O procedimento de medição de latência *one-way* das classes UDPPbrobe e TCPProbe é baseado em documentos do IPPM [2][3], onde é definido que cada amostra é obtida pela contagem do tempo entre a saída do pacote do *host* fonte e a chegada do mesmo no *host* destino. Esta métrica tem o nome Type-P-One-way-Delay e é composta pelos parâmetros: Src (endereço IP fonte), Dst (endereço IP destino) e T (tempo). Uma amostra de latência de um *host* fonte até um *host* destino no tempo T é denominada dT, isto é, o *host* fonte envia o pacote de teste no tempo T e o *host* destino recebe este pacote no tempo T+dT.

O procedimento de medição segue os seguintes passos:

- o sistema cliente no *host* fonte requisita um teste de latência ao sistema servidor no *host* destino da transmissão. As informações enviadas na requisição são a quantidade e o tamanho dos pacotes de teste que serão transmitidos;
- ambos *hosts* sincronizam seus relógios;
- no *host* fonte, o sistema cliente gera pacotes de teste com os endereços Src e Dst, inclui um identificador e o tempo atual nestes pacotes e os transmite para o *host* destino.
- no *host* destino, o sistema servidor aguarda o recebimento dos pacotes. Quando o pacote chega, o tempo de chegada e o tempo marcado no pacote são determinados o mais rápido possível. Uma estimativa da latência one-way pode ser calculada pela

subtração dos dois tempos. Se o pacote não chegar dentro de um tempo pré-determinado, a latência one-way é denominada indefinida.

- o valor de latência medido é colocado em um array de valores e, no final da medição, enviado de volta para o sistema cliente.

Nas medições de latência *one-way*, os pacotes devem chegar no destino dentro de um tempo determinado para não serem contados como perdidos. O valor da latência é medido em microsegundos. Esta métrica requer uma amostragem com frequência de duas vezes por segundo para que o tráfego de teste não perturbe o tráfego do usuário ou exceda suas reservas da rede.

4 Trabalhos Futuros

Esta biblioteca de classes continuará sendo aperfeiçoada para prover a maior funcionalidade e eficiência possível, sempre respeitando as especificações do grupo de trabalho IPPM. Ela faz parte de um projeto maior, em fase de final de implantação, denominado Mediator. Este sistema visa desenvolver um sistema de gerência de QoS para redes que operam com Serviços Diferenciados. Este sistema deverá realizar alocações de serviços diferenciados e monitoração da qualidade dos mesmos. Este projeto está sendo desenvolvido em parceria com a RMAV de Florianópolis.

5 Referências Bibliográficas

- [1] <http://www.ietf.org/html.charters/ippm-charter.html>
- [2] Paxson, V., Almes, G., Mahdavi, J. and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, Maio de 1998.
- [3] Almes, G., Kalidindi, S. and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, Setembro de 1999.