

Monitoramento de Tráfego de Rede com Psutil e Matplotlib

Abstract

Este trabalho descreve a implementação de um script para monitoramento de tráfego de rede em tempo real utilizando a biblioteca psutil para captura de dados de tráfego e a biblioteca matplotlib para a visualização desses dados em um gráfico. O script realiza a medição da largura de banda de uma interface de rede especificada, ao longo de um período determinado, e gera um gráfico que mostra a evolução do tráfego em Mbits/sec. Este tipo de monitoramento é útil para analisar o desempenho de redes em tempo real, fornecendo dados detalhados sobre a utilização da rede e permitindo a identificação de gargalos e falhas de desempenho.

1 Introdução

O monitoramento de tráfego de rede é uma tarefa crucial para a análise de desempenho de redes. Em um ambiente de rede, é importante monitorar constantemente a largura de banda para garantir que os dados estão sendo transmitidos de forma eficiente. Ferramentas como psutil permitem o acesso a informações detalhadas sobre o tráfego de rede, enquanto bibliotecas como matplotlib ajudam na visualização desses dados.

Este trabalho propõe um script que monitora a largura de banda de uma interface de rede durante um intervalo de tempo, capturando os dados de tráfego e gerando um gráfico com a evolução da largura de banda. O principal objetivo deste estudo é fornecer uma solução simples e eficaz para o monitoramento de tráfego em tempo real, com foco na análise de dados de rede para a otimização de sistemas e infraestrutura de rede.

Além disso, o monitoramento de redes é essencial para diagnosticar problemas de conectividade, congestionamento e desempenho em tempo real, especialmente em redes corporativas ou de grande porte, onde uma falha na infraestrutura pode causar sérios prejuízos.

2 Desenvolvimento / Proposta

O script desenvolvido consiste em duas funções principais: monitoramento e visualização dos dados coletados. A seguir, descrevemos o funcionamento de cada uma delas:

2.1 Monitoramento de Tráfego

A função `monitor_network_traffic` é responsável por monitorar o tráfego da rede. Utilizando a biblioteca `psutil`, ela acessa as estatísticas de tráfego da interface de rede fornecida (por padrão, `eth0`) a cada intervalo de tempo especificado (por padrão, 1 segundo) durante um período de 30 segundos.

A função captura a quantidade de bytes enviados e recebidos, calcula a largura de banda a cada intervalo e armazena os valores para posterior visualização. O tráfego é monitorado continuamente e, ao final do período, os valores de largura de banda são retornados.

A função de monitoramento se comunica diretamente com a interface de rede para obter as estatísticas de tráfego, o que a torna uma solução muito eficiente para medir o desempenho da rede. Este tipo de abordagem também pode ser facilmente adaptado para monitoramento em tempo real, onde os dados podem ser analisados imediatamente após a captura.

2.2 Visualização com Matplotlib

A função `plot_network_traffic` é responsável por gerar um gráfico utilizando a biblioteca `matplotlib`. Os valores de tempo e largura de banda, coletados pela função de monitoramento, são plotados em um gráfico de linha, onde o eixo X representa o tempo e o eixo Y a largura de banda em Mbits/sec. Esse gráfico permite a análise visual do desempenho da rede ao longo do tempo, facilitando a detecção de variações no tráfego e o comportamento da rede.

A visualização gráfica é importante porque permite que os usuários compreendam rapidamente a dinâmica do tráfego, observando momentos de pico, períodos de inatividade ou congestionamento. Com isso, é possível tomar decisões informadas sobre ajustes na rede para otimizar o desempenho.

2.3 Ajustes e Parâmetros

O script pode ser configurado para monitorar diferentes interfaces de rede, ajustar o tempo de monitoramento e o intervalo de captura de dados, conforme a necessidade do usuário. Os valores de largura de banda são coletados em unidades de Mbits/sec, o que facilita a interpretação dos dados.

Além disso, a implementação permite fácil personalização, como a escolha de diferentes intervalos de tempo e a definição de múltiplas interfaces para monitoramento simultâneo, tornando-o adequado para diferentes cenários de rede.

2.4 Impacto do Monitoramento Contínuo

A realização de monitoramento contínuo em redes pode ser crucial para a identificação precoce de problemas como congestionamentos, falhas de comunicação ou até mesmo ataques cibernéticos que afetam a performance da rede. Ferramentas como a proposta neste trabalho podem ser usadas para otimizar a utilização da rede, ajustando o tráfego conforme a necessidade ou baseando-se em análises preditivas de tráfego.

Por exemplo, um monitoramento constante pode permitir a detecção de variações no tráfego que indicam potenciais problemas, como pacotes sendo descartados devido a congestionamento ou pacotes com alta latência. Ao identificar esses problemas rapidamente, os administradores de rede podem agir de forma proativa para restaurar o desempenho ideal.

3 Avaliação de Desempenho

A avaliação de desempenho foi realizada executando o script em uma rede local. Durante o teste, monitoramos a interface de rede `eth0` durante um período de 30 segundos, com intervalos de 1 segundo entre as medições. A largura de banda foi capturada em tempo real e os dados foram plotados em um gráfico.

O gráfico abaixo mostra a evolução da largura de banda durante o monitoramento da rede.

Os resultados mostram que a largura de banda pode variar ao longo do tempo, devido a fatores como a carga de tráfego na rede, a quantidade de pacotes transmitidos e o congestionamento da rede. A função de monitoramento foi eficaz em capturar essas variações e gerar o gráfico correspondente.

3.1 Considerações sobre o Desempenho

Embora o script tenha sido eficiente para monitorar e visualizar a largura de banda, existem fatores externos que podem impactar a precisão das medições. A precisão das medições de largura de banda pode ser afetada por congestionamentos na rede, mudanças na carga de tráfego e até mesmo configurações de hardware que variam a velocidade de transmissão.

Além disso, ao realizar monitoramento em tempo real, é importante considerar a sobrecarga de processamento introduzida pela coleta constante de dados. Este

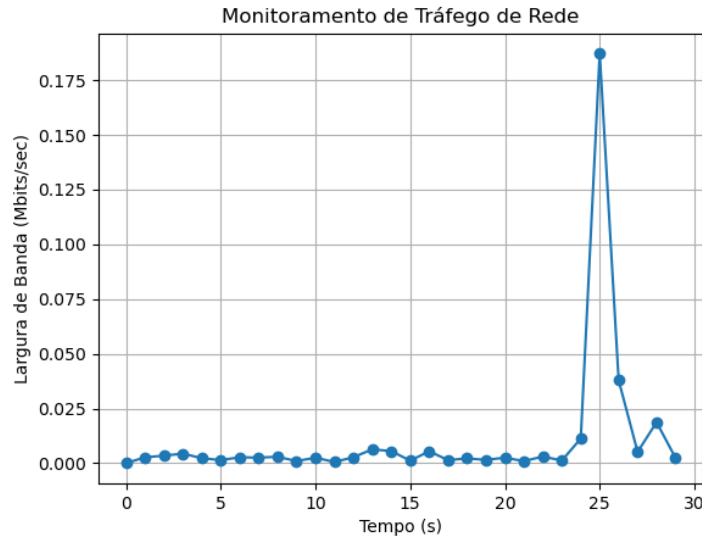


Figure 1: Gráfico de Largura de Banda em Tempo Real

impacto pode ser mitigado com a escolha de intervalos apropriados para coleta de dados e o uso de técnicas de amostragem.

3.2 Otimização do Monitoramento

Futuras implementações do script podem incluir a possibilidade de monitorar múltiplas interfaces simultaneamente ou a realização de testes em diferentes tipos de redes, como redes sem fio (Wi-Fi). A otimização de código também pode ser realizada, incluindo melhorias no cálculo de largura de banda em tempo real, considerando mais parâmetros de rede.

O uso de técnicas de processamento em tempo real e a integração de algoritmos de análise de tráfego pode aumentar significativamente o valor dos dados capturados, tornando-os mais relevantes para a análise de desempenho e a tomada de decisões.

4 Conclusão

A implementação do script para monitoramento de tráfego de rede utilizando `psutil` e visualização com `matplotlib` foi bem-sucedida. O gráfico gerado mostra claramente a variação na largura de banda durante o monitoramento da interface de rede. Este tipo de monitoramento pode ser usado para analisar o desempenho

de redes e otimizar sua infraestrutura, especialmente em redes corporativas ou de grande porte.

A solução proposta se mostrou simples, eficaz e de fácil implementação, podendo ser utilizada como uma ferramenta de monitoramento contínuo em diversas aplicações de redes. No entanto, algumas melhorias podem ser feitas para lidar com redes de maior escala ou para monitoramento de múltiplas interfaces simultaneamente.

5 Referências

- psutil. Python System and Process Utilities. Disponível em: <https://psutil.readthedocs.io/en/latest/>.
- matplotlib. Matplotlib: Python Plotting Library. Disponível em: <https://matplotlib.org/>.
- Tools and Methods for Network Performance Monitoring. Available at: <https://example.com>.