

# Documentação TP3

**Marcel Henrique S. Mendes**

<sup>1</sup>Departamento de Ciência da Computação (DCC) - Universidade Federal de Minas Gerais

`marcelmendes@dcc.ufmg.br`

## 1. Descrição geral

O terceiro trabalho prático da disciplina de redes consiste na implementação de um par cliente-servidor para recuperação de dados. A implementação é realizada utilizando chamadas de procedimento remoto (RPC). Foi usado o framework flask para a criação dos endpoints no servidor de acesso. A chamada de procedimento remoto é usada normalmente em ambientes de rede distribuídos, com a intenção de repassar a carga de processamento para um servidor remoto. A chamada normalmente é implementada por mensagens bem definidas entre as duas partes (cliente-servidor), no caso do trabalho implementado ela é realizada a partir do protocolo da camada de aplicação HTTP.

## 2. Decisões de implementação

No código do servidor foi implementado somente três endpoints, como pedido na especificação, cada um recuperando um tipo específico de dado. O primeiro endpoint recupera todos os dados dos Ixp's, o segundo recupera a lista de todas as redes associadas a um determinado Ixp e o terceiro o nome da rede identificada por net-id.

O desenvolvimento do código do cliente foi mais trabalhoso. A partir de uma requisição http ele recupera os dados dos endpoints. Foi criado um módulo chamado send-request que faz a requisição em todos os casos, basicamente passamos o cabeçalho de requisição GET como parâmetro e obtemos os dados desejados. Para cada requisição o cliente inicia uma nova conexão, apesar de não ser eficiente foi uma decisão tomada para facilitar o projeto.

Para cada endpoint foi criado um módulo para lidar com o recebimento das mensagens com o intuito de tratá-las adequadamente. Cada tipo de requisição tem suas peculiaridades em questão de deixar o dado preparado para a análise posterior. Após os dados estarem de forma organizada as funções de geração dos arquivos podem ser chamadas. Todo o dado necessário pode ser acessado através de três variáveis globais, definidas dessa forma para favorecer a simplicidade da implementação.

### 3. Análise

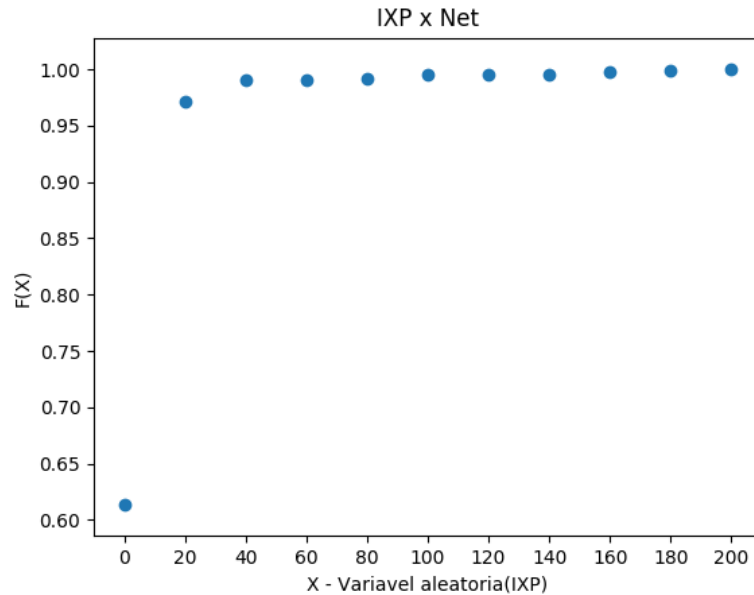


Figure 1. CDF- Análise zero

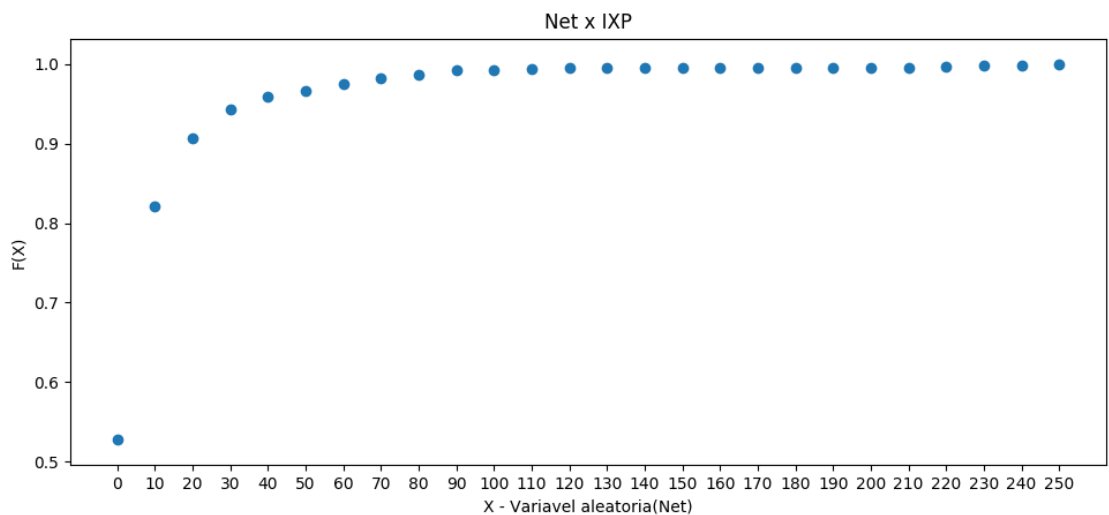


Figure 2. CDF- Análise um

Como podemos perceber no gráfico da análise 0 a probabilidade de uma rede qualquer (escolhida aleatoriamente do conjunto) não estar relacionada a nenhum Ixp é relativamente alta já que está próximo de 65% de chance. Podemos ver também que o número de redes relacionadas com menos de 20 IXP's também é alto. Uma quantidade pequena de IXP's concentram grande parte das redes.

O gráfico da análise 1 é um pouco mais uniforme, mas também nos mostra que a chance de escolhermos um IXP associado a somente uma rede é alta, próxima a 55%.

Como pode ser visto nos arquivos gerados pelo programa, o número de redes é maior que o número de IXP's por esse motivo os gráficos gerados são fiéis aos dados recolhidos. E como já previsto no primeiro gráfico, muitas redes estão associadas a um número pequeno de Pontos de troca de tráfego.

#### **4. Execução**

```
python3 server.py net.json ix.json netixlan.json  
python3 client.py ip:port ( 0 ou 1 )
```

#### **References**

1. *Distribution and Quantile Functions*, site [www.math.uah.edu](http://www.math.uah.edu), do Department of Mathematical Sciences da University of Alabama in Huntsville