

# **Engenharia de Tráfego Automatizada**

**Pesquisador:** Guilherme Almeida Andrade.

**Mentores/Especialistas:** Bernardo Falkoski, Flávio Silva, Renato Jucá, Vinícius Santolin, Hugo Barbosa.

## **O Projeto**

Automação de conexões manuais e alteração no fluxo do tráfego de rede permite melhorar questões relacionadas à configuração do tráfego de rede na Algar Telecom. Atualmente é necessário um profissional dedicado ao remanejamento do tráfego de rede, sendo que o mesmo deve conectar equipamento por equipamento para executar a configuração. Podemos tornar isso automático, impactando positivamente no capex e opex da empresa.

Desta forma, realizamos o estudo de tecnologias para entender o funcionamento de uma rede, verificando quais são suas características e algumas ferramentas que auxiliam em sua supervisão e manutenção. Foi verificado que as soluções presentes no mercado permitem a análise de dados, mas não a automação deste processo. A ideia do projeto é utilizar ferramentas como Arbor Sightline, plataforma de análise de dados que permite a análise de peering, verificar a capacidade de infraestrutura da rede e detectar ameaças, como suporte para a tomada de decisão de uma aplicação a ser desenvolvida, trabalhando com vários equipamentos diferentes.

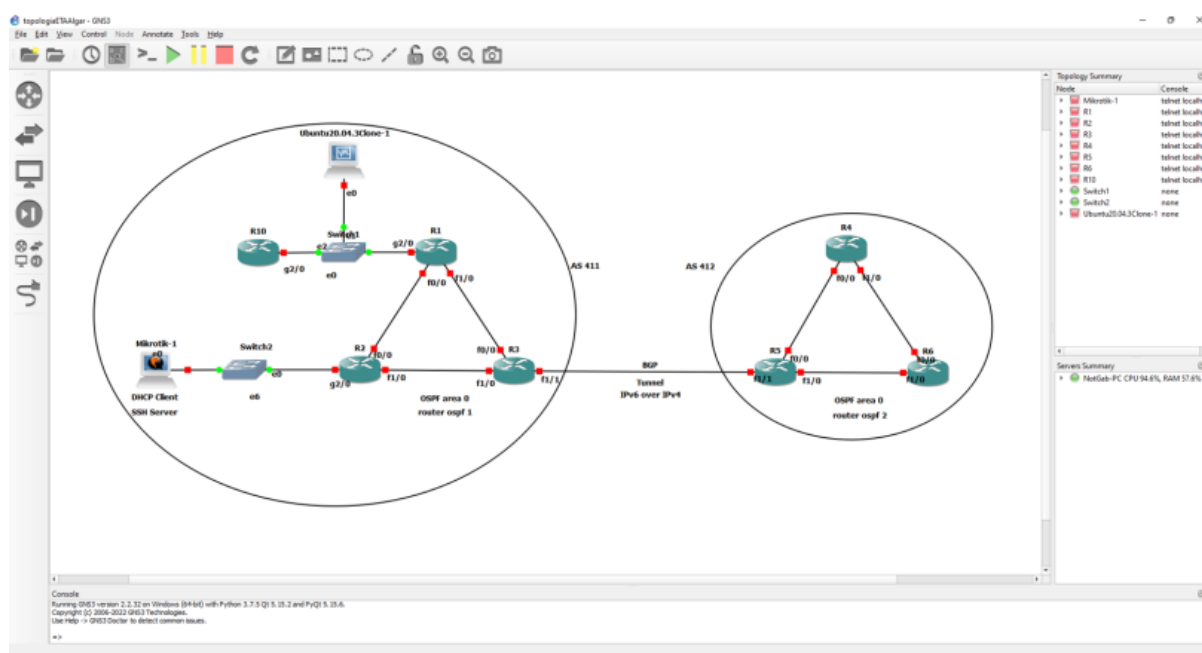
## **Desenvolvimento Inicial**

Como apontado, as soluções presentes no mercado permitem a análise de dados, mas não a automação deste processo. Portanto, inicialmente um programa em Python foi desenvolvido utilizando como base a biblioteca Netmiko para a automação de processos. Esta biblioteca simplifica a conexão CLI a equipamentos de rede, permitindo utilizar a conexão atualmente mais utilizada SSH. A premissa deste script é permitir a conexão em diversos equipamentos da rede, executar a configuração prevista para o mesmo, fazer o backup dos equipamentos da rede e, posteriormente, apresentar uma autonomia que não depende da ação de um profissional para realizar estas ações.

## Desenvolvimento da Automação

A topologia da Fig.(1) foi implementada no GNS3, um emulador de rede e para acesso dos equipamentos virtualizados, executado na máquina local. Os roteadores utilizados são Cisco e Mikrotik, mas o código implementado inicialmente automatiza somente as configurações de equipamentos Cisco. A fim de utilizar o código dentro do emulador é necessário adicionar uma máquina Linux à topologia.

Figura 1 - Topologia desenvolvida no GNS3.



O código completo pode ser observado no repositório do Github, disponível em [1]. A pasta "values" guarda arquivos com informações de conexão pré configurados, necessários para realizar o acesso aos equipamentos.

O arquivo *commands.json* guarda os parâmetros que podem ser utilizados na aplicação, separados por cada tipo de equipamento. Cada sequência de comandos possui: a ação que será realizada, os comandos que serão enviados ao equipamento e a indicação se este comando modifica algum dado do equipamento.

Para equipamentos Cisco temos:

- A ação "show network info", que possui os comandos:
  - "show ip interface brief", que exibe um resumo das interfaces em um dispositivo.
  - "show running-config", que mostra as configurações atuais do equipamento.

- A ação "sh run", que possui o comando:
  - "show ip interface brief", que exibe um resumo das interfaces em um dispositivo.
- A ação "set int ip", que possui os comandos:
  - "interface gigabitEthernet", que especifica o tipo de adaptador da porta e a localização da porta da interface configurada.
  - "ip address", seta o IP da interface.
  - "no shutdown", habilita a interface utilizada.
  - "do wr", salva as configurações atuais no equipamento.
- A ação " no set int ip", que possui o comando:
  - "interface gigabitEthernet", que especifica o tipo de adaptador da porta e a localização da porta da interface configurada.
  - "no ip address", remove o IP da interface.
  - "no shutdown", habilita a interface utilizada.
  - "do wr", salva as configurações atuais no equipamento.

Para equipamentos juniper temos:

- A ação "show network info", que possui os comandos:
  - "show ip interface brief", que exibe um resumos das interfaces em um dispositivo.
  - "show running-config", que mostra as configurações atuais do equipamento.

Todos os comandos podem ser encontrados na documentação dos fabricantes.

O arquivo *data.json* guarda as informações dos equipamentos que serão conectados. As informações armazenadas para cada equipamento são:

- "hostname" - Nome do host.
- "device\_type" - Tipo do equipamento
- "host" - IP do host.
- "port" - Porta do equipamento.
- "username" - Usuário.
- "password" - Senha.
- "executable" - Em execução ou não.

Na raiz temos o arquivo principal, com a interface de usuário desenvolvida, onde é possível escolher as ações que serão realizadas. Por meio desta interface é possível conectar a um host e realizar os comandos definidos pelo usuário, previamente listados. As ações realizadas geram logs completos que mostram qual equipamento foi o alvo da operação, se houve sucesso ou fracasso na operação realizada, apresentando a data completa de realização.

Desta forma, foi desenvolvida uma ferramenta que facilita a conexão e configuração de múltiplos equipamentos que ocupam a topologia desenvolvida, executando diversos comandos. É uma solução simples, de fácil manutenção e suporte, obtendo assim, mais tempo para a equipe atuar estrategicamente.

### **Ampliação da Automação**

Após verificar o funcionamento do código no emulador GNS3, resolveu-se ampliar a ferramenta para utilização em laboratório utilizando o emulador Eve-NG. Dentre os motivos que motivaram a evolução para esta ferramenta pode-se citar:

- Software multivendor.
- Máquina Linux não é necessária na topologia
- Clientless, onde o acesso às topologias é realizado através de um navegador, por qualquer usuário com acesso a rede.
- Multiusuário, onde várias pessoas têm acesso aos diversos laboratórios.
- Interação com rede real.
- Instâncias de lab simultâneas.
- Transposição rápida de uma rede real para virtualizada.

Com isto em mente, os laboratórios iniciais foram desenvolvidos utilizando o Eve-NG. A Fig.(2) mostra um exemplo de topologia desenvolvido, onde três AS foram emulados e uma conexão BGP foi realizada entre eles, como pode ser observada na Fig.(3), que apresenta a configuração do roteador R5. Estas configurações foram realizadas em segundos por meio de um script Python, possibilitando também a definição de outras interfaces nos roteadores, interface de loopback e outras configurações de equipamento. Para realizar esta operação é necessário somente uma conexão com a rede onde a topologia está emulada, apresentada em 100.100.100.0.

Figura 2 - Topologia BGP desenvolvida no EVE-NG.

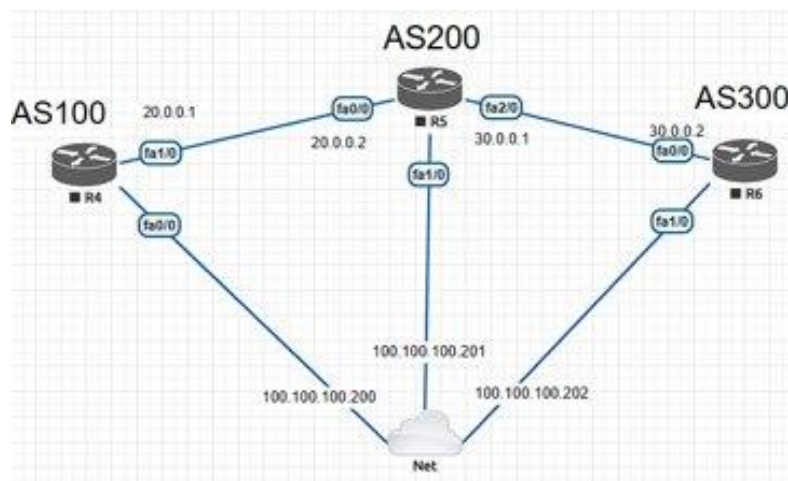


Figura 3 - Configurações “show bgp” do roteador R5.

```
R5
For address family: IPv4 Unicast
BGP neighbor is 20.0.0.1, remote AS 100, external link
BGP version 4, remote router ID 1.1.1.1
BGP state = Established, up for 00:00:21
Last read 00:00:21, last write 00:00:21, hold time is 180, keepalive interval
is 60 seconds
Neighbor sessions:
  1 active, is not multisession capable (disabled)
Neighbor capabilities:
  Route refresh: advertised and received(new)
  Four-octets ASM Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
  Enhanced Refresh Capability: advertised and received
  Multisession Capability:
  Stateful switchover support enabled: NO for session 1
Message statistics:
  InQ depth is 0
  OutQ depth is 0

      Sent      Rcvd
Opens:          1         1
Notifications:  0         0
Updates:        0         0
--More--
```

Em outros laboratórios, foram utilizadas imagens de diversos *vendors*, realizando suas configurações tanto manualmente, quanto através de códigos Python, verificando o funcionamento e compatibilidade do código desenvolvido, confirmando a PoC.

## Business Case: Benchmarking Nokia

Finalizada a PoC em que verificou-se o funcionamento do projeto e possibilidade de ampliação para uma aplicação robusta, o business case tornou-se necessário para observar os

impactos positivos desta automação em uma implementação real. Deste modo, um benchmarking de um estudo montado pela Nokia foi realizado.

Neste estudo, disponível em [2], a empresa Analysys Mason foi responsável por pesquisar os benefícios da automação de redes IP. O projeto coletou mais de 60 pontos de dados de estratégias de automação da rede das operadoras de cinco operadoras diferentes, para desenvolver um modelo de benefícios dimensionado para uma operadora regional de grande escala. Os processos de automação foram divididos em três categorias: Execução de Serviços, Manutenção do Ciclo de Vida da Rede e Garantia de Rede e Serviços. Após uma análise, foi verificado que o foco da Engenharia de Tráfego e seu Troubleshooting está nas duas últimas categorias.

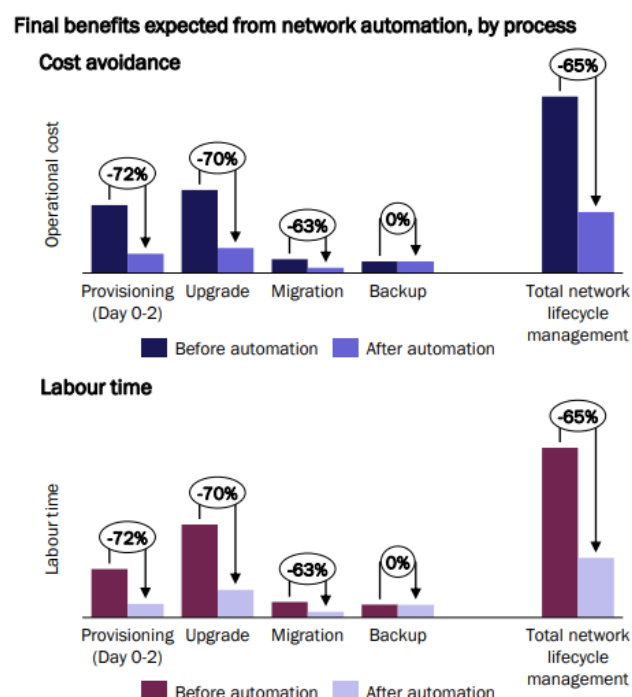
Entende-se por Manutenção do Ciclo de Vida da Rede o provisionamento, configuração e manutenção de equipamentos de rede, após a instalação física do hardware. Isso inclui 4 processos distintos: provisionamento, backup, atualização e migração de serviço.

Desta categoria, verificou-se:

- -72% do custo e tempo de trabalho do Provisionamento
- -70% do custo e do tempo de trabalho de Melhoria
- -63% do custo e tempo de trabalho de Migração
- Total de 65% de Redução de Custos e Tempo de Trabalho

Estes dados podem ser observados na Fig.(4).

Figura 4 - Benefícios esperados da automação de rede - Manutenção do Ciclo de Vida.



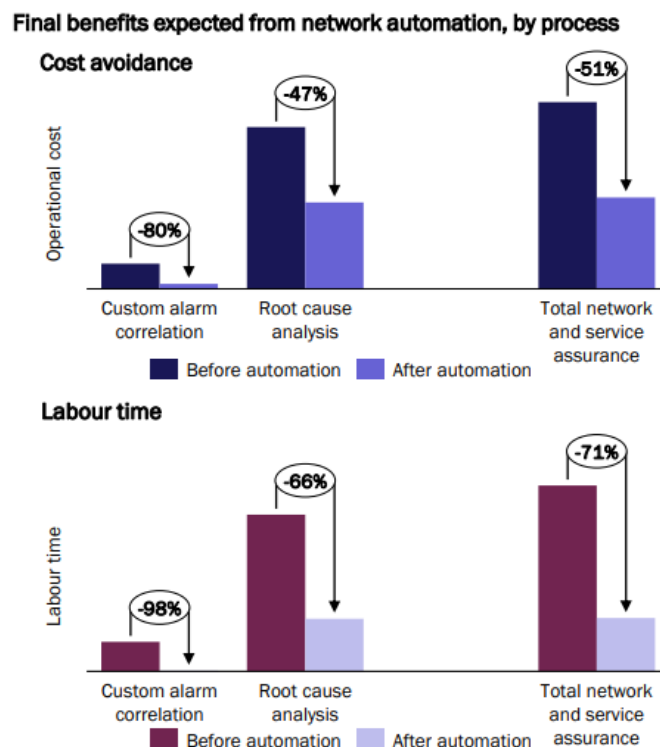
Entende-se por Garantia de Rede e Serviços o gerenciamento de falhas com correlação de alarmes e análise de causa raiz: a correlação de alarme personalizada inclui os esforços necessários para identificar e correlacionar falhas e alarmes a um problema comum na rede. A análise de causa raiz refere-se aos esforços necessários para identificar e resolver a principal causa de uma interrupção do serviço ou degradação do desempenho.

Desta categoria, verificou-se:

- -80% do custo de Correlação de Alarmes
- -47% do custo de Análise da Raiz de Problemas
- -98% do tempo de Correlação de Alarmes
- -66% do tempo de Análise da Raiz de Problemas
- Total de 51% de Redução de Custos e 71% do Tempo de Trabalho

Estes dados podem ser observados na Fig.(5).

Figura 5 - Benefícios esperados da automação de rede - Garantia de Rede e Serviço



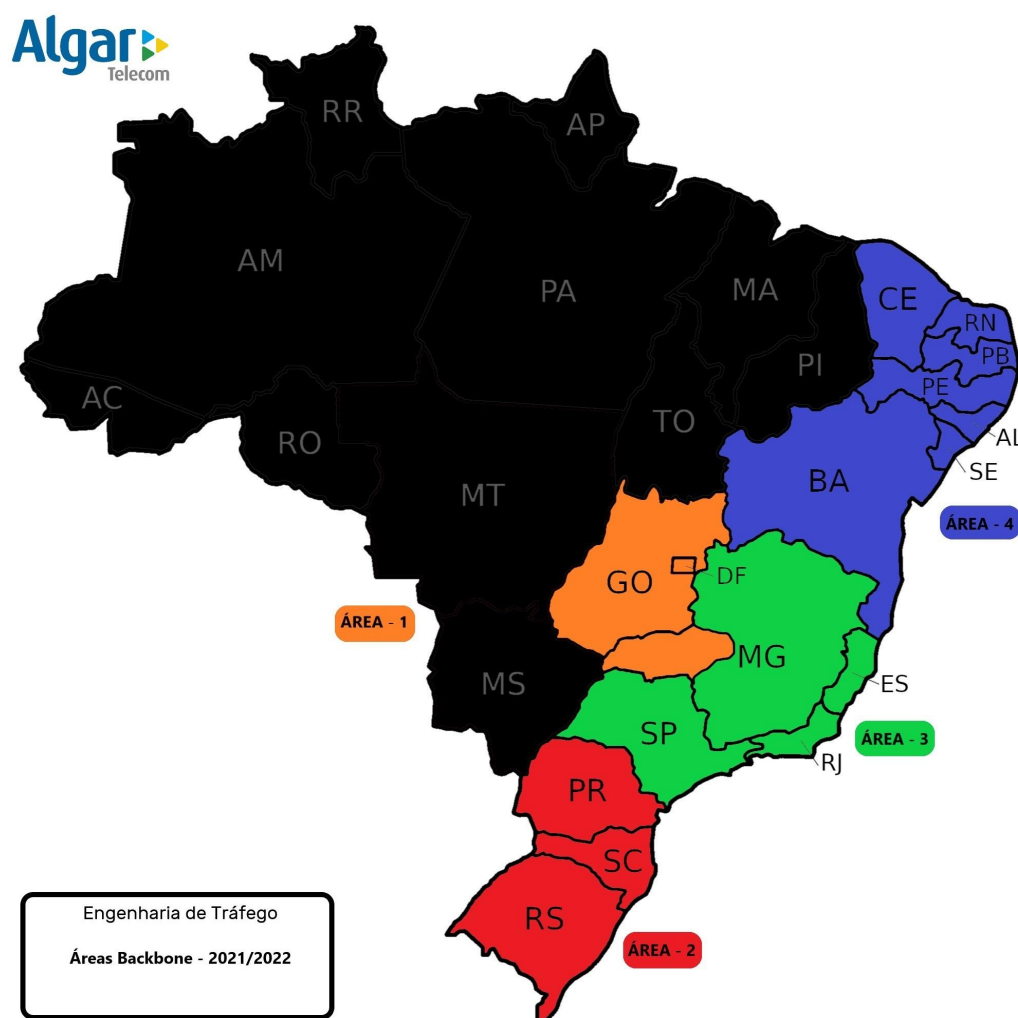
Dentro do contexto de automação apresentado, quatro cases foram apresentados com premissas de automação que se encaixam naquelas utilizadas no estudo realizado pela Nokia, a fim de utilizar os processos de automação de redes/tráfego em um novo projeto.

## Business Case: Casos de Teste

- **Configuração de Escopo**

A fim de realizar a configuração de escopo o engenheiro seleciona a porta lógica/física do equipamento e realiza a conexão da porta física até o DGO. Por meio da utilização de transceivers é realizado um peering entre operadoras e este peering é separado em regiões, como pode ser observado na Fig.(6).

Figura 6 - Área de Backbone Algar Telecom.



Levantar uma configuração de escopo leva em média 45min a 3hrs atualmente e, como pode ser observado na Fig.(7), são utilizados comandos padronizados que alteram em algumas informações específicas para cada configuração como AS, IP etc. A ideia deste case seria modelar um sistema genérico de preenchimento destas



informações específicas, que junto a um script Python levantaria a configuração em segundos.

Figura 7 - Configuração de escopo Vivo.

```
vsantolin@c-bb-bor-pae-002-mx204-01> show configuration policy-options policy-statement Export_PNI_Vivo |display set
set policy-options policy-statement Export_PNI_Vivo term Route_Security from policy Route_Security
set policy-options policy-statement Export_PNI_Vivo term Route_Security then reject
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_NO_Export from community CTBC_Customer_No_Export
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_NO_Export then reject
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community BRAS-Regiao-ULA
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community BRAS-Regiao-PMS
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community BRAS-Regiao-IUA
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community BRAS-Regiao-GNA
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community BRAS-Regiao-URA
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community BRAS-Regiao-FRU
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community Algar_Customer_Area1_I1ha11
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community Algar_Customer_Area1_I1ha12
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community Algar_Customer_Area1_I1ha10
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community Algar_Customer_Area1_I1ha18
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community Algar_Customer_Area1_I1ha19
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 from community Algar_Customer_Area1_I1ha20
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 then metric 30
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area1 then next term
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area2 from community Algar_Customer_Area2_I1ha1
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area2 from community Algar_Customer_Area2_I1ha2
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area2 from community Algar_Customer_Area2_I1ha3
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area2 from community Algar_Customer_Area2_I1ha4
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area2 from community BRAS-Regiao-PAE
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area2 then metric 10
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area2 then next term
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community Algar_Customer_Area3_I1ha7
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community Algar_Customer_Area3_I1ha8
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community Algar_Customer_Area3_I1ha9
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community Algar_Customer_Area3_I1ha13
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community Algar_Customer_Area3_I1ha14
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community Algar_Customer_Area3_I1ha15
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community Algar_Customer_Area3_I1ha16
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community Algar_Customer_Area3_I1ha17
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community BRAS-Regiao-BHE
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community BRAS-Regiao-SMG
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 from community BRAS-Regiao-SPO
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 then metric 20
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area3 then next term
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 from community Algar_Customer_Area4_I1ha5
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 from community Algar_Customer_Area4_I1ha6
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 from community BRAS-Regiao-AJU
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 from community BRAS-Regiao-FLA
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 from community BRAS-Regiao-JPA
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 from community BRAS-Regiao-MCO
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 from community BRAS-Regiao-RCE
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 from community BRAS-Regiao-SDR
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 then metric 40
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export_Area4 then next term
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Prepnd_1 from community CTBC_Prepnd_Nac_1
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Prepnd_1 then as-path-prepend 16735
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Prepnd_1 then accept
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Prepnd_2 from community CTBC_Prepnd_Nac_2
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Prepnd_2 then as-path-prepend "16735 16735"
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Prepnd_2 then accept
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Prepnd_3 from community CTBC_Prepnd_Nac_3
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Prepnd_3 then as-path-prepend "16735 16735 16735"
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Prepnd_3 then accept
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export from community CTBC_Customer_BGP
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Community_Export then accept
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Partial_Transit from community Customer_Partial_Transit
set policy-options policy-statement Export_PNI_Vivo term BGP_Customer_Partial_Transit then accept
set policy-options policy-statement Export_PNI_Vivo term Export_CTBC_Networks from community Rotas_CTBC
set policy-options policy-statement Export_PNI_Vivo term Export_CTBC_Networks then accept
set policy-options policy-statement Export_PNI_Vivo term Reject then reject
```

Ainda dentro deste contexto de peers entre operadoras separados em regiões, tem-se como ideia automatizar a manipulação de tráfego regional de Peerings e OTTs com a utilização de código em Python, com o objetivo de diminuir trabalhos manuais. Para esta tarefa, têm-se como premissas:

1. Considerar equipamentos da linha Cisco e Juniper que façam roteamento dinâmico;
2. Utilização de plataformas existentes (Arbor Sightline, NorthStar) que deem o insight para as localidades que precisem de alguma atuação do tráfego;
3. Case Nordeste: Regionalização de 2Gbps de tráfego originário no Nordeste sentido São Paulo que foi confinado na região Nordeste.

A última premissa representa um bom exemplo de como uma atuação automatizada no tráfego seria de grande ajuda para os engenheiros de rede. A rede estruturante da Algar e suas localidades possui alguns caches de conteúdo, como Facebook, Google, Akamai, Netflix,

PSN, acessados pelos clientes através da internet. Recentemente na região Nordeste havia um comportamento anômalo, mesmo havendo o cache de conteúdo em Salvador, os clientes desta área estavam utilizando a eletrônica da Bahia até São Paulo a fim de buscar um conteúdo já disponível no Nordeste. Utilizando dos recursos de engenharia de tráfego, como a manipulação de algumas rotas BGP, os engenheiros de tráfego conseguiram o tratamento deste comportamento anômalo. Após as modificações, a interface que apresentava apenas 200Mbps de tráfego subiu para 2Gbps, como pode ser observado na Fig.(8).

Figura 8 - Variação da Interface após modificações nas rotas BGP.

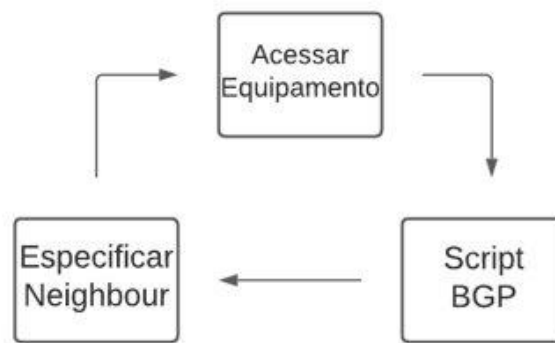


Ao considerar este circuito estatístico, foi calculado um valor de R\$70/Mb, ou cerca de R\$120.000 de custo salvo em eletrônica por meio deste PIA-F confinado em Fortaleza.

- **Subir Sessão BGP**

A conexão BGP é responsável por conectar as redes, esta configuração é muito comum no dia a dia de um engenheiro de rede, como estudado e verificada em laboratórios no Eve-NG, ela é realizada em um passo a passo formado comandos padronizados, simplificados na Fig.(9). Portanto, a ideia deste case é automatizar atributos do BGP em um script a fim de salvar tempo e esforço.

Figura 9 - Passo a passo BGP.



- **Troubleshooting de Rede**

Sabe-se que isolar e identificar um problema de rede pode levar horas e geralmente são utilizados comandos para realizar o diagnóstico e encontrar a raiz do problema. Desta forma, têm-se como ideia criar uma rotina de comandos padrão como ping, tracert, para verificar a conexão com equipamentos e a rota para esta conexão, conectar esta rotina automatizada um banco de dados, e assim comparar com o comportamento padrão da rede, identificando de forma eficiente os problemas. Esta rotina deve guardar logs que facilitam a solução de problemas futuros.

- **Verificação de Equipamentos**

Por fim, este case defende a ideia da criação de um BOT/Script que faça a varredura em todos os equipamentos a fim de levantar a configuração de elementos que manipulam tráfego. Deste modo, seria possível identificar equipamentos que possuem alguma otimização de configuração como backdoor e lixo de configuração de circuitos já desativados. Para esta implementação, as seguintes premissas são utilizadas:

1. Considerar equipamentos homologados pela Algar Telecom;
2. Criação de uma rotina semanal de varredura de protocolos ativos indevidamente, como o protocolo telnet, portas com status down há um tempo determinado, a fim de dar sobrevida ao equipamento e permitir a otimização de seus recursos.

A última premissa baseia-se em um problema enfrentado pelo time de ativação de clientes. Em vários momentos, novas interfaces são necessárias para realizar a ativação de novos clientes e em vários casos não existe nenhuma disponível. Ao realizar uma verificação em

vários equipamentos, o que demanda tempo, várias interfaces down que não haviam sido registradas eram identificadas. Erros como esse geram custos a empresa, que muitas vezes precisa realizar a aquisição de novos equipamentos por não identificar disponibilidade naqueles já adquiridos. Identificar estes problemas demanda esforço e tempo que pode variar de uma a três horas, a depender do número de configurações do equipamento.

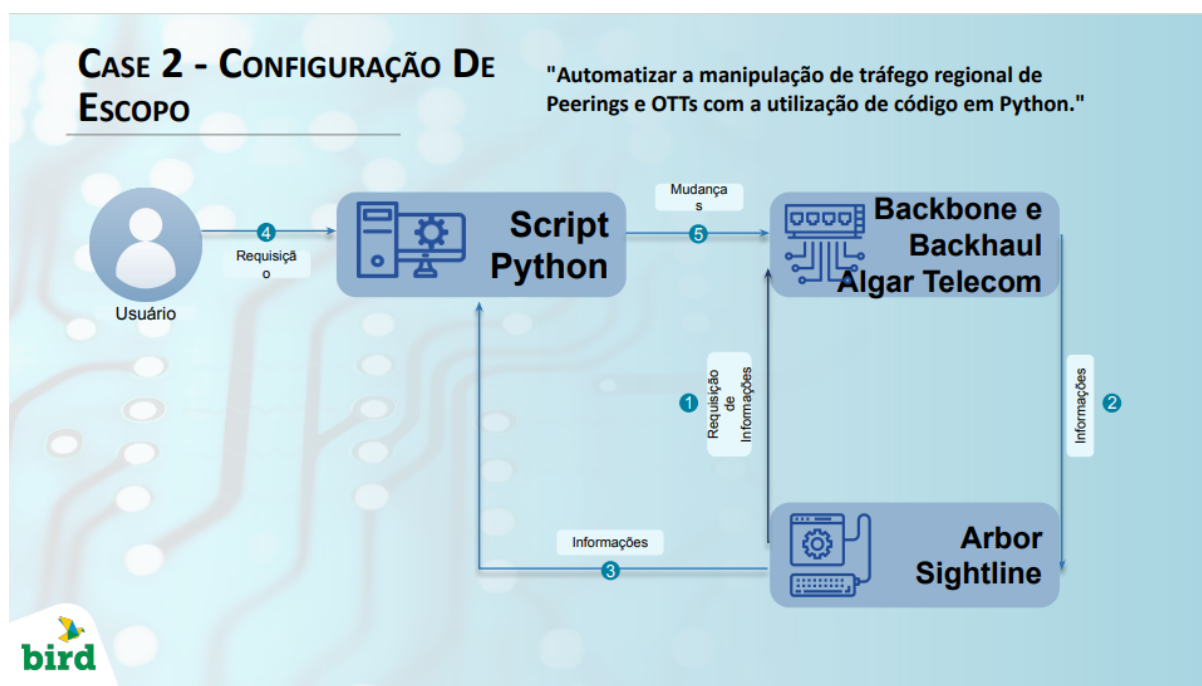
### Evolução do Projeto Para Uma Nova PoC

Ao realizar a análise dos cases apresentados junto ao time da operação, ativação de clientes e engenharia de tráfego, a Verificação de Equipamentos e Configuração de Escopo são aqueles que se destacam para a ampliação à uma nova PoC. Seus diagramas de funcionamento estão, respectivamente, apresentados nas Fig.(10) e Fig.(11).

Figura 10 - Diagrama Verificação de Equipamentos



Figura 11 - Diagrama Configuração de Escopo



## Referências

[1] Algar Telecom FIC (2022). Repositório Github.

<https://github.com/algar-telecom-fic/AutomacaoETA>. Acesso em: 22 fev. 2023.

[2] THE QUANTITATIVE benefits of IP network automation: A study from Analysys Mason based on operator data. [S. l.], 2021. Disponível em:

<https://www.nokia.com/networks/automation/ip-and-optical-network-automation/measurable-benefits/>. Acesso em: 26 jan. 2023.