

Topologia de rede com utilização de protocolos TCP/IP

GBC066 - Arquitetura de Redes TCP/IP

Gabriel Ribeiro Bernardi
Guilherme Soares Correa

gabrielrbernardi@gmail.com, guilhermescorrea2014@gmail.com

02-Agosto, 2022

Bacharelado em Ciência da Computação
Universidade Federal de Uberlândia, Uberlândia, Minas Gerais

1 Resumo

Este trabalho tem como objetivo explorar os conceitos vistos durante a disciplina de Arquitetura de Redes TCP/IP, no curso de ciência da computação. Serão abordadas as configurações de equipamentos virtualizados, utilizando o simulador GNS3 e protocolos e serviços, como: BGP, OSPF, DHCP, SSH, entre outros.

2 Introdução

Em suma, a ideia do projeto é estudar, entender e replicar alguns conceitos que podem ser vistos em um núcleo de rede (backbone), com a utilização de diversos protocolos de roteamento (estáticos e dinâmicos) somados com as configurações de equipamentos e serviços de diferentes fabricantes e Sistemas Operacionais (SO).

Esse protótipo de artigo traz alguns pontos importantes relacionados às configurações de protocolos e serviços, assim como o detalhamento da criação da topologia de rede, com seus respectivos equipamentos e algumas amostras do funcionamento dos mesmos.

Além disso, serão apresentadas algumas amostras da utilização e configuração do GNS3 (Graphical Network Simulator-3) e seus respectivos equipamentos.

Como exemplificação do uso das ferramentas utilizadas neste projeto, tem-se como amostra o núcleo da rede de um Provedor de Serviços de Internet (ISP), onde os mesmos fazem o uso de protocolos de roteamento dentro das áreas, com suas configurações, além de protocolos de roteamento para comunicação entre os ASs e a utilização de serviços para conexão aos equipamentos a fim de configurar, modificar e/ou consertar configurações de rede nos mesmos.

3 Estado da arte / Considerações iniciais

Para o desenvolvimento do projeto, foi necessário entender o princípio de funcionamento de alguns protocolos e serviços, que serão mostrados a seguir.

3.1 AS

Resumidamente, é uma área da rede onde os equipamentos presentes estão conectados utilizando prefixos de rede IP (Internet Protocol), sob o controle de um ou mais operadores de rede¹. Cada AS possui um valor identificador, chamado de ASN (Autonomous System Number).

3.2 BGP

O BGP (Border Gateway Protocol) é um protocolo de roteamento de rotas utilizado para fazer a comunicação de dados para a rede externa, via comunicação entre Sistemas Autônomos². Para fazer a utilização desses sistemas, o BGP implementa o algoritmo de roteamento hierárquico, tendo como vantagem que no mundo real, nos diferentes ASs não são utilizados o mesmo protocolo de roteamento, tendo como vantagem a utilização do BGP. Com essa forma de comunicação pode-se dizer também que o BGP é um protocolo de roteamento exterior (EGP).

3.3 OSPF

O OSPF (Open Shortest Path First) é um protocolo de roteamento de redes IPs, onde implementa o algoritmo de roteamento de estado de enlace, onde, na prática, um determinado equipamento possui conhecimento da rota e custo do enlace somente para equipamentos vizinhos³. Além disso, o OSPF faz a utilização do roteamento presente dentro de um determinado AS, conhecido como Intra-AS e pode ganhar a classificação de IGP, ou protocolo de roteamento interno.

3.4 DHCP

O DHCP (Dynamic Host Configuration Protocol) é um protocolo de gerenciamento de redes IPs responsável por atribuir automaticamente endereços de rede para as máquinas conectadas que solicitam o serviço⁴. O protocolo é considerado como um serviço da camada de rede

¹Cloud Flare, 2022.

²Fortinet, 2022.

³Cisco, sshamim, rvigil, 2013.

⁴Microsoft, 2022.

e é composto por quatro partes, para o fornecimento do endereço. De forma simplificada, primeiramente, a máquina que deseja usar o serviço faz o broadcast de uma mensagem (DISCOVERY) a fim de descobrir o servidor que está executando o DHCP. Após isso, o servidor retorna uma mensagem dizendo que está disponível para atender a solicitação (OFFER). Em seguida, o cliente faz a solicitação de um endereço de rede (REQUEST) e esse servidor, após entender a solicitação, retorna uma mensagem de conhecimento (ACKNOWLEDGMENT) e em seguida informa ao cliente o endereço de IP para navegar na rede.

3.5 SSH

O SSH (Secure Shell Protocol) é um protocolo de rede criptográfico, considerado um serviço da camada de aplicação, capaz de oferecer uma conexão segura por meio de uma rede insegura para conexão e gerenciamento de um determinado equipamento⁵. A arquitetura do SSH é composta por um cliente, que deseja fazer a conexão, e um servidor, que é aquele que será acessado.

3.6 Wireshark

O software Wireshark é uma ferramenta muito importante para poder analisar os pacotes que trafegam pela rede⁶. Por meio dele foi possível observar se os protocolos estavam funcionando corretamente, se os serviços estavam executando e se a comunicação estava de fato ocorrendo. As capturas de pacotes efetuadas para demonstração foram guardadas para posterior análise.

3.7 Equipamentos e Plataformas

Para a virtualização da topologia de rede foi utilizado o software GNS3, capaz de fazer a virtualização de redes complexas, compostas por equipamentos de diferentes marcas e SOs. Além disso, foram múltiplas instâncias virtualizadas do roteador Cisco 7200 e Mikrotik (6.48.6 Long-term, Raw Disk Image) além da utilização de uma máquina virtual executando Ubuntu 20.04.3 e equipamentos nativos e instâncias de VPCs (Virtual PC), capaz de executar um SO simples e leve mas suficiente para testar algumas conexões e protocolos. Por fim,

⁵IETF RFC 4251, 2022.

⁶Wireshark, 2022.

foram utilizados alguns Switches, capazes de ”multiplicar” uma determinada interface de um equipamento.

4 Metodologia

A metodologia foi dividida em algumas partes, sendo elas: criação da topologia de rede e definição de serviços a serem utilizados; Definição das redes a serem implementadas; Início da virtualização de rede e configuração dos protocolos e interfaces dos equipamentos; Captura de pacotes e amostra do funcionamento das etapas;

4.1 Topologia, protocolos e serviços utilizados

A rede é composta por 16 equipamentos e a topologia geral pode ser vista na Figura 1. Para roteamento interno, foi escolhido o OSPF, roteamento InterAS, o BGP, sendo que a conexão InterAS é feita utilizando IPv6. Para isso foi necessário fazer um túnel, onde é feito o encapsulamento dos dados IPv4 para IPv6 e após a saída do túnel, o desencapsulamento dos mesmos. Além disso, foi necessário configurar rotas estáticas para o direcionamento e redistribuição de rotas de diferentes protocolos entre os Sistemas Autônomos. Como serviço da camada de aplicação, foi escolhido o SSH e para serviço da camada de rede, o DHCP.

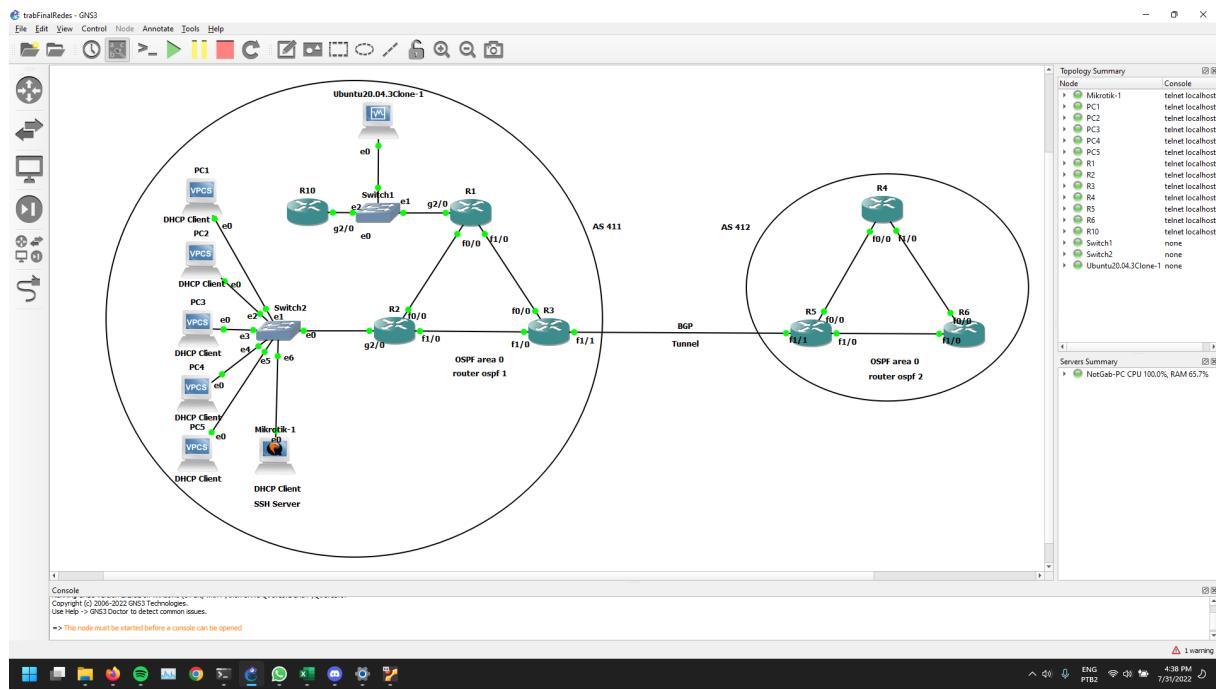


Figura 1: Topologia de rede geral

Tabela 1: Listagem dos IPs destinados às interfaces e equipamentos.

Início da tabela			
Equipamento	Interfaces	IP	Observação
R1	f0/0	192.168.1.1	
R1	f1/0	192.168.2.1	
R1	g2/0	192.168.20.3	
R2	f0/0	192.168.1.2	
R2	f1/0	192.168.3.1	
R2	g2/0	192.168.30.1	DHCP Server
R3	f0/0	192.168.2.2	
R3	f1/0	192.168.3.2	
R3	f1/1	2022:ABC:DB1::1	Máscara /64
R3	tunnel0	192.168.9.1	
R3	loopback1	172.168.20.5	Interface de Loopback do OSPF

Continuação da tabela 1			
Equipamento	Interfaces	IP	Observação
R10	g2/0	192.168.20.4	
R3	tunnel0	192.168.20.5	Máquina Ubuntu 20.04.3
R4	f0/0	192.168.4.1	
R4	f1/0	192.168.5.1	
R5	f0/0	192.168.4.2	
R5	f1/0	192.168.6.1	
R5	f1/1	2022:ABC:DB1::2	Máscara /64
R5	tunnel0	192.168.9.2	
R5	loopback1	172.0.0.2	
R6	f0/0	192.168.5.2	
R6	f1/0	192.168.6.2	
Mikrotik	ether0	DHCP	DHCP Client
PC1	ether0	DHCP	DHCP Client
PC2	ether0	DHCP	DHCP Client
PC3	ether0	DHCP	DHCP Client
PC4	ether0	DHCP	DHCP Client
PC5	ether0	DHCP	DHCP Client
Swtich 1	Ethernet 0, 1 e 2	&	
Switch 2	Ethernet 0, 1, 2, 3, 4, 5 e 6	&	
Fim da tabela			

4.2 AS 412

Para configuração do AS 412, três roteadores (R4, R5 e R6) Cisco 7200 utilizam, para comunicação entre si, a interface fastEthernet, com protocolo de roteamento OSPF, na área 0, e o protocolo IPv4, para os endereços de rede.

No roteador R5, considerado como roteador de borda, algumas configurações tiveram que ser diferentes quando comparado com as configurações dos demais equipamentos do AS 412. Nesse equipamento foi ativado o IPv6, para passagem dos dados para o AS vizinho. Como dentro do AS estava sendo utilizado o IPv4, foi necessário criar um túnel, que faz o encapsulamento dos pacotes IPv4 para IPv6 e assim possam ser transportados por ele. O protocolo de roteamento configurado foi o BGP. A interface de saída de R5 (f1/1) possuí dois endereços IPs. Um para identificação da interface pelo outro equipamento no AS vizinho e um endereço para identificação do túnel, para passagem dos dados. Também foi necessário fazer a redistribuição das rotas OSPF⁷ para o outro AS. Tal configuração é feita a partir da configuração do BGP. Por fim, foi necessário criar um roteamento estático para direcionar as rotas do AS vizinho para o endereço de IP do túnel vizinho.

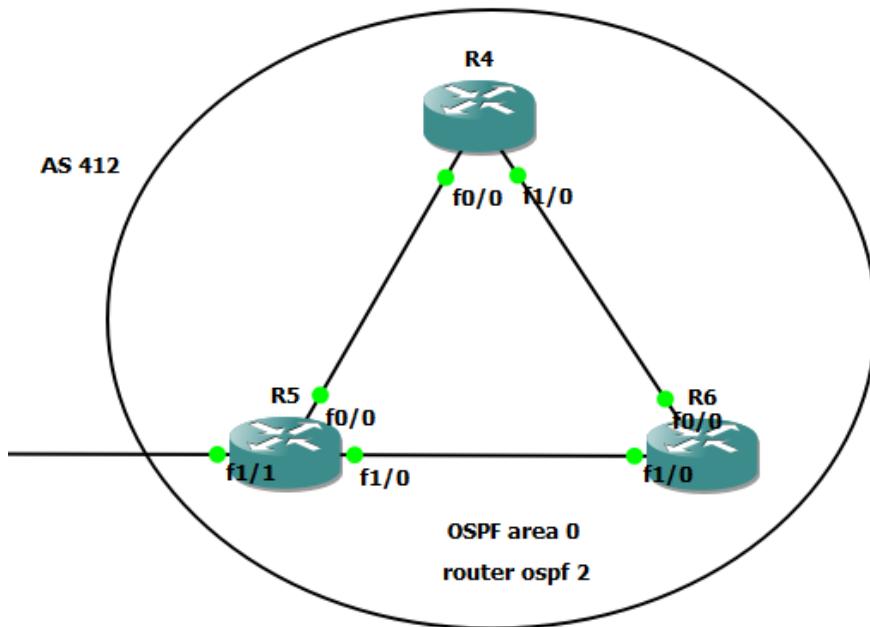


Figura 2: Fração da topologia, com foco no AS 412

4.3 AS 411

A composição do AS 411 tem a utilização de protocolos de roteamento juntamente com a utilização de serviços da camada de rede e de aplicação, além de endereços IPs da versão

⁷Must Be Geek, Arranda Saputra, 2019.

4. A topologia parcial é composta por 13 equipamentos, de diversos fornecedores, sendo quatro dispositivos Cisco 7200, um Mikrotik, uma Máquina Virtual (VM) para conexão aos equipamentos, possuindo um sistema mais robusto, cinco dispositivos VPCs e dois Switches.

4.3.1 Núcleo do AS

Para roteamento entre os roteadores Cisco (R1, R2 e R3), fez-se a utilização do protocolo de roteamento OSPF, com área 0. A interface g2/0 tem configuração para possuir uma interface de repasse de informações, onde fazer com que R10 e a VM utilizem a interface para conexão com a rede.

4.3.2 A máquina virtual

A utilização da máquina virtual tornou-se bastante importante durante o período de teste e configuração do serviço de camada de aplicação escolhido, o SSH. Com a utilização da mesma foi possível executar os testes de conexão, capturar os pacotes trafegados na rede além de ser uma interface mais fácil de configurar, em relação a rede. A VM possui configuração para utilizar 2GB de memória RAM, dois núcleos de CPU, CoreI5 9300H e 256MB de memória de vídeo e Sistema Operacional Ubuntu 20.04.3 LTS, com virtualização via VirtualBox. Após a criação e configuração da mesma, a VM foi importada para o GNS3 e assim foi possível efetuar a conexão dela com o restante da rede. Com essa importação, a máquina foi conectada em um enlace, com gateway de saída para a interface g2/0, de R1. A configuração de rede pode ser vista na Figura 3.

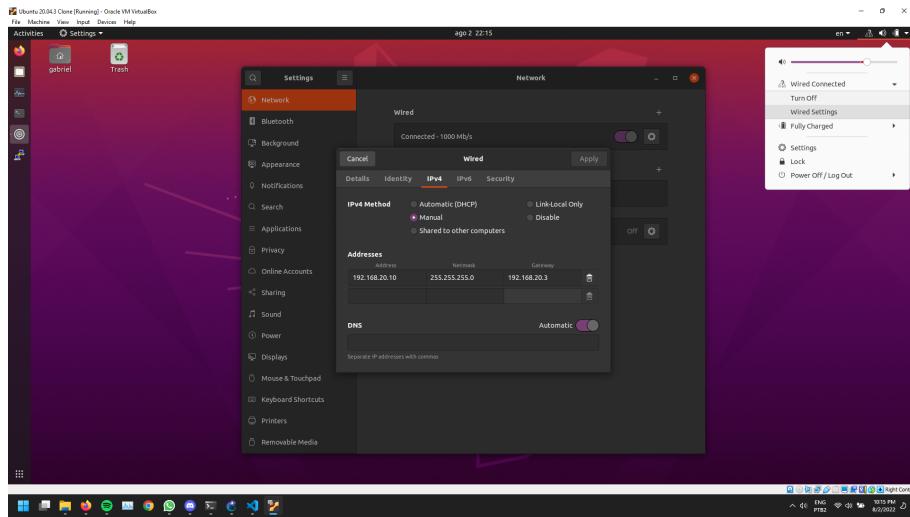


Figura 3: Configuração de rede da Máquina Virtual

4.3.3 A utilização do DHCP

Como serviço da camada de rede, foi utilizado o DHCP, responsável por distribuir endereços de IP dinâmicos para os equipamentos conectados ao servidor. O roteador R3 foi responsável por executar esse serviço, que estava disponível a partir da interface g2/0 e a mesma estava conectada a um Switch. Além disso, ao Switch estavam conectados seis equipamentos, sendo cinco VPCs, máquinas simples para testes de serviços e protocolos, e um roteador Mikrotik.

4.3.4 O serviço da camada de aplicação

O serviço escolhido para a camada de aplicação foi o SSH, que possibilita a conexão remota a equipamentos para configuração, acesso a arquivos, entre outros. Inicialmente, foi tentado utilizar um roteador Cisco para fazer a hospedagem do servidor SSH, no entanto, foi posteriormente escolhido fazer o uso em um roteador Mikrotik. Diante disso, o roteador Mikrotik estava responsável por utilizar o DHCP para receber um IP da rede e diante disso, fazer o uso do servidor SSH e disponibilizá-lo para a rede.

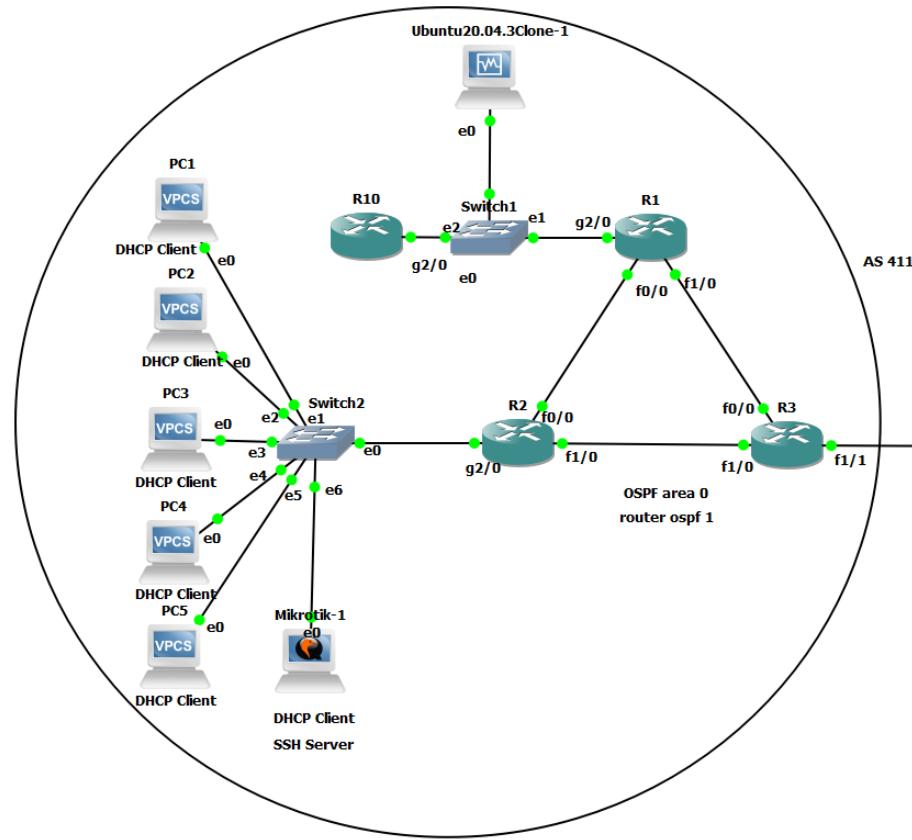


Figura 4: Fração da topologia, com foco no AS 411

4.4 Conexão InterAS

Mais especificamente, foram utilizados dois equipamentos, Cisco 7200, para a conexão e troca de dados entre os ASs configurados. Os roteadores R3 e R5 estavam conectados por meio de interfaces fastEthernet 1/1, onde ambas faziam a implementação de um túnel, para fazer o encapsulamento de pacotes IPs da versão 4 para pacotes IPs da versão 6. Além disso, ambos equipamentos possuíam, em suas interfaces voltadas para a rede externa (InterAS), dois endereços IPs cada, um IPv6, para identificar as mesmas e fazer a transferência de informações entre os ASs, e um endereço IPv4, para a identificação das interfaces de rede pelos demais equipamentos da área.

Por fim, foi necessário também fazer a configuração de redistribuição de rotas entre os Sistemas Autônomos, para que o dispositivo em uma região pudesse comunicar com outro dispositivo em outro AS.

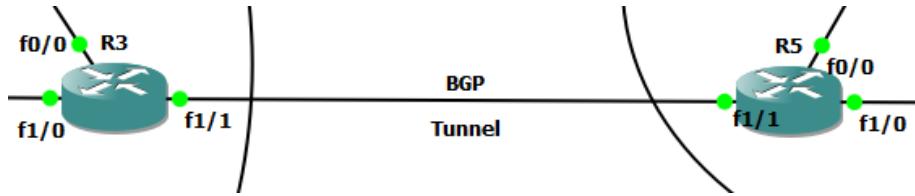


Figura 5: Fração da topologia, com foco na conexão InterAS

5 Experimentos e resultados

Após a devida configuração dos equipamentos, os serviços foram testados, por meio de ferramentas como ping e traceroute, e para verificação, os pacotes trafegados na rede foram capturados utilizando o software Wireshark.

5.1 O protocolo OSPF

Para os testes do protocolo OSPF foram executados dois experimentos, um de conexão em equipamentos dentro do AS 411 e outro em equipamentos do AS 411 com origem ao AS 412.

5.1.1 Roteamento InterAS (AS 411)

Para verificar se os pacotes e o roteamento estão acontecendo de forma correta nos equipamentos foi executado o ping de um roteador em outro. No caso do exemplo da Figura 5, tem-se a execução e a captura de pacotes (Figura 6) trafegados entre R1 e R2. Além disso, o Wireshark foi configurado para capturar os pacotes que estavam trafegando pelo enlace que conectava R1 em R2.

```
R1#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/42/56 ms
R1#
```

Figura 6: Execução do ping de R1 para R2

6 8.650478 192.168.1.1	192.168.1.2	ICMP	114 Echo (ping) request id=0x0001, seq=0/0, ttl=255 (reply in 8)
7 0.000000 ca:02:1f:88:00:00	ca:02:1f:88:00:00	LOOP	60 Reply
8 0.021941 192.168.1.2	192.168.1.1	ICMP	114 Echo (ping) reply id=0x0001, seq=0/0, ttl=255 (request in 6)
9 0.020454 192.168.1.1	192.168.1.2	ICMP	114 Echo (ping) request id=0x0001, seq=1/256, ttl=255 (reply in 10)
10 0.010972 192.168.1.2	192.168.1.1	ICMP	114 Echo (ping) reply id=0x0001, seq=1/256, ttl=255 (request in 9)
11 0.045876 192.168.1.1	192.168.1.2	ICMP	114 Echo (ping) request id=0x0001, seq=2/512, ttl=255 (reply in 12)
12 0.001995 192.168.1.2	192.168.1.1	ICMP	114 Echo (ping) reply id=0x0001, seq=2/512, ttl=255 (request in 11)
13 0.010970 192.168.1.1	192.168.1.2	ICMP	114 Echo (ping) request id=0x0001, seq=3/768, ttl=255 (reply in 14)
14 0.004987 192.168.1.2	192.168.1.1	ICMP	114 Echo (ping) reply id=0x0001, seq=3/768, ttl=255 (request in 13)
15 0.017461 192.168.1.1	192.168.1.2	ICMP	114 Echo (ping) request id=0x0001, seq=4/1024, ttl=255 (reply in 16)
16 0.013963 192.168.1.2	192.168.1.1	ICMP	114 Echo (ping) reply id=0x0001, seq=4/1024, ttl=255 (request in 15)

Figura 7: Captura de pacotes via Wireshark, no ping de R1 para R2

5.1.2 Redistribuição de rotas InterAS (AS 411 para AS 412)

Outro teste efetuado foi em relação a redistribuição de rotas do AS 411 para o AS 412. Para isso, também foi executado o ping e a captura de pacotes foi feita, via Wireshark. Além disso, a ferramenta traceroute foi utilizada para que pudesse ser possível observar os saltos que os pacotes estavam fazendo até chegar ao destino. O teste efetuado foi de o roteador R1 verificar se R4 está ativo (Figura 7), além se saber os saltos que foram dados (Figura 8). Ambos os testes funcionaram corretamente, demonstrando que a redistribuição de rotas do OSPF⁸ estava acontecendo como esperado. Para os experimentos, o Wireshark foi configurado para capturar os pacotes trafegados pelo túnel de ligação, entre R3 e R5.

```
R1#ping 192.168.5.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 148/172/196 ms
R1#traceroute 192.168.5.1
Type escape sequence to abort.
Tracing the route to 192.168.5.1
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.2.2 72 msec 48 msec 56 msec
  2 192.168.9.2 124 msec 132 msec 168 msec
  3 192.168.4.1 184 msec
    192.168.6.2 160 msec
    192.168.4.1 192 msec
R1#
```

Figura 8: Execução do ping e traceroute, de R1 para R4

7 5.476026 192.168.2.1	192.168.5.1	ICMP	158 Echo (ping) request id=0x0000, seq=0/0, ttl=254 (reply in 8)
8 0.057845 192.168.5.1	192.168.2.1	ICMP	158 Echo (ping) reply id=0x0000, seq=0/0, ttl=254 (request in 7)
9 0.000757 192.168.2.1	192.168.5.1	ICMP	158 Echo (ping) request id=0x0000, seq=1/256, ttl=254 (reply in 10)
10 0.105718 192.168.5.1	192.168.2.1	ICMP	158 Echo (ping) reply id=0x0000, seq=1/256, ttl=254 (request in 9)
11 0.001834 192.168.2.1	192.168.5.1	ICMP	158 Echo (ping) request id=0x0000, seq=2/512, ttl=254 (reply in 12)
12 0.079787 192.168.5.1	192.168.2.1	ICMP	158 Echo (ping) reply id=0x0000, seq=2/512, ttl=254 (request in 11)
13 0.000757 192.168.2.1	192.168.5.1	ICMP	158 Echo (ping) request id=0x0000, seq=3/768, ttl=254 (reply in 14)
14 0.109707 192.168.5.1	192.168.2.1	ICMP	158 Echo (ping) reply id=0x0000, seq=3/768, ttl=254 (request in 13)
15 0.102725 192.168.2.1	192.168.5.1	ICMP	158 Echo (ping) request id=0x0000, seq=4/1024, ttl=254 (reply in 16)
16 0.069813 192.168.5.1	192.168.2.1	ICMP	158 Echo (ping) reply id=0x0000, seq=4/1024, ttl=254 (request in 15)

Figura 9: Captura de pacotes via Wireshark, no ping de R1 para R4

⁸Must Be Geek, Arranda Saputra, 2019.

19 0.871343 192.168.2.1	192.168.5.1	UDP	86 49157 → 33437 Len=0
20 0.055853 192.168.9.2	192.168.2.1	ICMP	114 Time-to-live exceeded (Time to live exceeded in transit)
21 0.098244 192.168.2.1	192.168.5.1	UDP	86 49158 → 33438 Len=0
22 0.032912 192.168.9.2	192.168.2.1	ICMP	114 Time-to-live exceeded (Time to live exceeded in transit)
23 0.104229 192.168.2.1	192.168.5.1	UDP	86 49159 → 33439 Len=0
24 0.039403 192.168.9.2	192.168.2.1	ICMP	114 Time-to-live exceeded (Time to live exceeded in transit)
25 0.107222 192.168.2.1	192.168.5.1	UDP	86 49160 → 33440 Len=0
26 0.102725 192.168.4.1	192.168.2.1	ICMP	114 Destination unreachable (Port unreachable)
27 0.071808 192.168.2.1	192.168.5.1	UDP	86 49161 → 33441 Len=0
28 0.116688 192.168.6.2	192.168.2.1	ICMP	114 Time-to-live exceeded (Time to live exceeded in transit)
29 0.082778 192.168.2.1	192.168.5.1	UDP	86 49162 → 33442 Len=0
30 0.071808 192.168.4.1	192.168.2.1	ICMP	114 Destination unreachable (Port unreachable)

Figura 10: Captura de pacotes via Wireshark, no traceroute de R1 para R4

5.2 Túnel, protocolo BGP, IPv6 e redistribuição de rotas

Para o teste do túnel, assim como para verificar se o protocolo BGP estava funcionando corretamente, foram efetuados dois experimentos.

5.2.1 Primeiro experimento

O primeiro experimento foi dividido em duas partes, uma somente com a utilização de IPv6 e outra para testar a conexão do túnel.

5.2.1.1 Experimento utilizando IPv6

A primeira parte do primeiro experimento foi de efetuar o ping e traceroute entre os equipamentos R3 e R5, assim como a captura dos pacotes trafegados, mediante execução dos comandos, para atestar que o enlace que conecta os dois equipamentos estavam funcionando corretamente.

```
R3#ping 2022:ABC:DB1::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2022:ABC:DB1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/22/36 ms
R3#traceroute 2022:ABC:DB1::2
Type escape sequence to abort.
Tracing the route to 2022:ABC:DB1::2

 1 2022:ABC:DB1::2 16 msec 44 msec 32 msec
R3#
```

Figura 11: Execução do ping e traceroute, de R3 para R5, com IPv6

4 0.311674 2022:abc:db1::1	2022:abc:db1::2	ICMPv6	114 Echo (ping) request id=0x1188, seq=0, hop limit=64 (reply in 5)
5 0.015958 2022:abc:db1::2	2022:abc:db1::1	ICMPv6	114 Echo (ping) reply id=0x1188, seq=0, hop limit=64 (request in 4)
6 0.014961 2022:abc:db1::1	2022:abc:db1::2	ICMPv6	114 Echo (ping) request id=0x1188, seq=1, hop limit=64 (reply in 7)
7 0.015958 2022:abc:db1::2	2022:abc:db1::1	ICMPv6	114 Echo (ping) reply id=0x1188, seq=1, hop limit=64 (request in 6)
8 0.014960 2022:abc:db1::1	2022:abc:db1::2	ICMPv6	114 Echo (ping) request id=0x1188, seq=2, hop limit=64 (reply in 9)
9 0.016954 2022:abc:db1::2	2022:abc:db1::1	ICMPv6	114 Echo (ping) reply id=0x1188, seq=2, hop limit=64 (request in 8)
10 0.014461 2022:abc:db1::1	2022:abc:db1::2	ICMPv6	114 Echo (ping) request id=0x1188, seq=3, hop limit=64 (reply in 11)
11 0.014961 2022:abc:db1::2	2022:abc:db1::1	ICMPv6	114 Echo (ping) reply id=0x1188, seq=3, hop limit=64 (request in 10)
12 0.000997 2022:abc:db1::1	2022:abc:db1::2	ICMPv6	114 Echo (ping) request id=0x1188, seq=4, hop limit=64 (reply in 13)
13 0.014959 2022:abc:db1::2	2022:abc:db1::1	ICMPv6	114 Echo (ping) reply id=0x1188, seq=4, hop limit=64 (request in 12)

Figura 12: Captura de pacotes via Wireshark, no ping de R3 para R5

16 0.880924 2022:abc:db1::1	2022:abc:db1::2	UDP	62 50153 → 33434 Len=0
17 0.014961 2022:abc:db1::2	2022:abc:db1::1	ICMPv6	110 Destination Unreachable (Port unreachable)
18 0.015956 2022:abc:db1::1	2022:abc:db1::2	UDP	62 62684 → 33435 Len=0
19 0.015958 2022:abc:db1::2	2022:abc:db1::1	ICMPv6	110 Destination Unreachable (Port unreachable)
20 0.013962 2022:abc:db1::1	2022:abc:db1::2	UDP	62 58129 → 33436 Len=0
21 0.025441 2022:abc:db1::2	2022:abc:db1::1	ICMPv6	110 Destination Unreachable (Port unreachable)

Figura 13: Captura de pacotes via Wireshark, no traceroute de R3 para R5

5.2.1.2 O funcionamento do túnel

Para a verificação do funcionamento do túnel, foi efetuado, também, o ping e o traceroute entre R3 e R5. No entanto, o destino era a interface de saída do túnel, com IPv4. Além disso, o experimento 5.1.2 também mostra que o túnel estava funcionando corretamente.

```
R3#ping 192.168.9.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.9.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/77/108 ms
R3#traceroute 192.168.9.2
Type escape sequence to abort.
Tracing the route to 192.168.9.2
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.9.2 28 msec 40 msec 48 msec
R3#
```

Figura 14: Execução do ping e traceroute, de R3 para R5, via túnel

380 0.676223 192.168.9.1	192.168.9.2	ICMP	158 Echo (ping) request id=0x0000, seq=0/0, ttl=255 (reply in 381)
381 0.034907 192.168.9.2	192.168.9.1	ICMP	158 Echo (ping) reply id=0x0000, seq=0/0, ttl=255 (request in 380)
382 0.032912 192.168.9.1	192.168.9.2	ICMP	158 Echo (ping) request id=0x0000, seq=1/256, ttl=255 (reply in 383)
383 0.033910 192.168.9.2	192.168.9.1	ICMP	158 Echo (ping) reply id=0x0000, seq=1/256, ttl=255 (request in 382)
384 0.033908 192.168.9.1	192.168.9.2	ICMP	158 Echo (ping) request id=0x0000, seq=2/512, ttl=255 (reply in 385)
385 0.042885 192.168.9.2	192.168.9.1	ICMP	158 Echo (ping) reply id=0x0000, seq=2/512, ttl=255 (request in 384)
386 0.025931 192.168.9.1	192.168.9.2	ICMP	158 Echo (ping) request id=0x0000, seq=3/768, ttl=255 (reply in 387)
387 0.045877 192.168.9.2	192.168.9.1	ICMP	158 Echo (ping) reply id=0x0000, seq=3/768, ttl=255 (request in 386)
388 0.025931 192.168.9.1	192.168.9.2	ICMP	158 Echo (ping) request id=0x0000, seq=4/1024, ttl=255 (reply in 389)
389 0.033908 192.168.9.2	192.168.9.1	ICMP	158 Echo (ping) reply id=0x0000, seq=4/1024, ttl=255 (request in 388)

Figura 15: Captura de pacotes via Wireshark, no ping de R3 para R5, via túnel

390 6.168363 192.168.9.1	192.168.9.2	UDP	86 49154 → 33434 Len=0
391 0.021941 192.168.9.2	192.168.9.1	ICMP	114 Destination unreachable (Port unreachable)
392 0.008976 192.168.9.1	192.168.9.2	UDP	86 49155 → 33435 Len=0
393 0.034907 192.168.9.2	192.168.9.1	ICMP	114 Destination unreachable (Port unreachable)
394 0.026927 192.168.9.1	192.168.9.2	UDP	86 49156 → 33436 Len=0
395 0.016956 192.168.9.2	192.168.9.1	ICMP	114 Destination unreachable (Port unreachable)

Figura 16: Captura de pacotes via Wireshark, no traceroute de R3 para R5, via túnel

5.2.2 Segundo experimento

O segundo teste foi feito para efetuar também o ping e traceroute, mas entre R1 para R4. O experimento feito no item 5.1.2 atesta que o túnel, o protocolo BGP e a redistribuição de rotas entre os Sistemas Autônomos estava funcionando corretamente.

5.3 O serviço DHCP

Para a verificação da utilização correta do serviço DHCP, foi utilizado um roteador Cisco 7200, um Switch, cinco VPCs e um roteador Mikrotik (MK). O MK e os VPCs fazem o uso do IP dinâmico, fornecido pelo servidor DHCP, para conexão e transferência de dados pela rede. O roteador R2 está atuando como servidor. Para captura dos pacotes trafegados relacionados ao DHCP, foi colocado o Wireshark no enlace entre R2 e o Switch 2. Com isso, foi possível observar as trocas de mensagens para recebimento de endereço via DHCP.

```
PC4> ip dhcp
DDORA IP 192.168.30.3/24 GW 192.168.30.1
PC4> █
```

Figura 17: Configuração de VPC para requisição e recebimento de endereço IP via DHCP

17 3.692464 0.0.0.0		255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x25171dbd
18 0.085771 ca:02:1f:88:00:38	Broadcast	ARP	60 Who has 192.168.30.2? Tell 192.168.30.1	
19 0.920204 0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x25171dbd	
20 1.025465 192.168.30.1	255.255.255.255	DHCP	342 DHCP Offer - Transaction ID 0x25171dbd	
21 0.029920 0.0.0.0	255.255.255.255	DHCP	342 DHCP Request - Transaction ID 0x25171dbd	
22 0.016954 192.168.30.1	255.255.255.255	DHCP	342 DHCP ACK - Transaction ID 0x25171dbd	
35 1.126942 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0x1fa0113e	
36 0.012969 ca:02:1f:88:00:38	Broadcast	ARP	60 Who has 192.168.30.3? Tell 192.168.30.1	
37 0.750082 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0x4c946316	
38 0.014959 ca:02:1f:88:00:38	Broadcast	ARP	60 Who has 192.168.30.4? Tell 192.168.30.1	
39 0.224906 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0x1fa0113e	
40 0.660183 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0x7988b56e	
41 0.014962 ca:02:1f:88:00:38	Broadcast	ARP	60 Who has 192.168.30.5? Tell 192.168.30.1	
42 0.089048 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0x4c946316	
43 0.254642 192.168.30.1	192.168.30.3	DHCP	342 DHCP Offer - Transaction ID 0x1fa0113e	
44 0.586782 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0xa77c0747	
45 0.014959 ca:02:1f:88:00:38	Broadcast	ARP	60 Who has 192.168.30.6? Tell 192.168.30.1	
46 0.045385 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0x7988b56e	
47 0.119680 192.168.30.1	192.168.30.4	DHCP	342 DHCP Offer - Transaction ID 0x4c946316	
48 0.733395 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0xa77c0747	
50 0.074798 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0xa77c0747	
51 0.074800 192.168.30.1	192.168.30.5	DHCP	342 DHCP Offer - Transaction ID 0x7988b56e	
52 0.329120 0.0.0.0	255.255.255.255	DHCP	406 DHCP Request - Transaction ID 0x1fa0113e	
53 0.014961 192.168.30.1	192.168.30.3	DHCP	342 DHCP ACK - Transaction ID 0x1fa0113e	
54 0.494721 0.0.0.0	255.255.255.255	DHCP	406 DHCP Discover - Transaction ID 0xa77c0747	
55 0.104720 192.168.30.1	192.168.30.6	DHCP	342 DHCP Offer - Transaction ID 0xa77c0747	
56 0.149599 0.0.0.0	255.255.255.255	DHCP	406 DHCP Request - Transaction ID 0x4c946316	
57 0.014960 192.168.30.1	192.168.30.4	DHCP	342 DHCP ACK - Transaction ID 0x4c946316	
58 0.224399 Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.30.3 (Request)	
59 0.523600 192.168.30.1	192.168.30.7	DHCP	342 DHCP Offer - Transaction ID 0xa77c0747	
60 0.135549 0.0.0.0	255.255.255.255	DHCP	406 DHCP Request - Transaction ID 0x7988b56e	
61 0.014471 192.168.30.1	192.168.30.5	DHCP	342 DHCP ACK - Transaction ID 0x7988b56e	
62 0.089762 Private_66:68:01	Broadcast	ARP	64 Gratuitous ARP for 192.168.30.4 (Request)	
63 0.239385 Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.30.3 (Request)	
64 0.598843 0.0.0.0	255.255.255.255	DHCP	406 DHCP Request - Transaction ID 0xa77c0747	
65 0.014984 192.168.30.1	192.168.30.6	DHCP	342 DHCP ACK - Transaction ID 0xa77c0747	
68 0.071841 192.168.30.2	255.255.255.255	MNDP	155 5678 → 5678 Len=113	
69 0.014927 0c:40:6e:cc:00:00	CDP/VTP/DTP/PagP/UD...	CDP	107 Device ID: RouterOS Port ID: ether1	
70 0.002994 0c:40:6e:cc:00:00	LLDP_Multicast	LLDP	110 MA/0c:40:6e:cc:00:00 IN/ether1 120 SysN=Rc	
71 0.150177 Private_66:68:00	Broadcast	ARP	64 Gratuitous ARP for 192.168.30.3 (Request)	
72 0.510375 0.0.0.0	255.255.255.255	DHCP	406 DHCP Request - Transaction ID 0xa77c0747	
73 0.014962 192.168.30.1	192.168.30.7	DHCP	342 DHCP ACK - Transaction ID 0xa77c0747	

Figura 18: Trechos de captura de pacotes via Wireshark, para solicitação de endereço IP via DHCP

5.4 Conexão remota SSH

Para a realização deste experimento, inicialmente foram utilizados três roteadores Cisco 7200, dois Switches, uma Máquina Virtual e um roteador Mikrotik. Inicialmente, foi detectado que nos roteadores Cisco, o serviço SSH é desabilitado por padrão, para o caso do servidor. Para cliente, o serviço é habilitado por padrão. No entanto, para habilitar, é necessário que sejam feitas algumas configurações. Foram seguidos diversos passo a passos disponibilizados na internet, muitos deles pela própria Cisco⁹, além de alguns outros fornecidos por blogs¹⁰¹¹,

⁹Cisco, 2015.

¹⁰Solutions, 2020.

¹¹Auvik, Kevin Dooley, 2014.

fóruns¹²¹³¹⁴¹⁵¹⁶, e vídeos no Youtube¹⁷.

5.4.1 Conexão entre dois roteadores Cisco

Para essa etapa, o roteador R1 foi utilizado como servidor SSH e R10 como cliente. No entanto não foi possível estabelecer uma conexão, devido a alguma barreira presente no R1.

5.4.2 Conexão entre VM e R1

Para a segunda parte do experimento, foi utilizada a Máquina Virtual, citada anteriormente, juntamente com o roteador R1. Novamente foi tentada estabelecer uma conexão entre os dois equipamentos, mas não foi possível estabelecer a conexão. Além disso, o papel de cliente e servidor foi trocado por ambos diversas vezes. Ademais, um switch foi adicionado para multiplicação da interface de rede disponível por R1, para que R10 e a VM pudessem tentar a conexão em R1. Durante uma das tentativas de conexão da VM (cliente) e R1 (servidor), foram capturados pacotes, mostrando que estava tentando estabelecer uma conexão entre os dois, mas por algum motivo, a conexão estava sendo encerrada.

4 4.080956 192.168.20.10	192.168.20.3	TCP	74 37938 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1895533173 TSecr=0 WS=128
5 0.0004881 192.168.20.3	192.168.20.10	TCP	60 22 → 37938 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=1460
6 0.000976 192.168.20.10	192.168.20.3	TCP	60 37938 → 22 [ACK] Seq=1 Ack=1 Win=64240 Len=0
7 0.000000 192.168.20.10	192.168.20.3	SSHv2	95 Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.5)
8 0.005759 192.168.20.3	192.168.20.10	SSHv2	73 Server: Protocol (SSH-2.0-Cisco-1.25)
9 0.000000 192.168.20.10	192.168.20.3	TCP	60 37938 → 22 [ACK] Seq=42 Ack=20 Win=64221 Len=0
10 0.000977 192.168.20.10	192.168.20.3	TCP	1514 37938 → 22 [ACK] Seq=42 Ack=20 Win=64221 Len=1460 [TCP segment of a reassembled PDU]
11 0.000000 192.168.20.10	192.168.20.3	SSHv2	106 Client: Key Exchange Init
12 0.0009748 192.168.20.3	192.168.20.10	TCP	60 22 → 37938 [ACK] Seq=20 Ack=1554 Win=4076 Len=0
13 0.000000 192.168.20.3	192.168.20.10	SSHv2	398 Server: Key Exchange Init
14 0.000000 192.168.20.10	192.168.20.3	TCP	60 37938 → 22 [ACK] Seq=1554 Ack=364 Win=63877 Len=0
15 0.000977 192.168.20.10	192.168.20.3	TCP	60 37938 → 22 [FIN, ACK] Seq=1554 Ack=364 Win=63877 Len=0
16 0.005759 192.168.20.3	192.168.20.10	TCP	60 22 → 37938 [ACK] Seq=364 Ack=1555 Win=4076 Len=0
17 0.085887 192.168.20.3	192.168.20.10	TCP	60 22 → 37938 [FIN, PSH, ACK] Seq=364 Ack=1555 Win=4076 Len=0
18 0.000000 192.168.20.10	192.168.20.3	TCP	60 37938 → 22 [ACK] Seq=1555 Ack=365 Win=63877 Len=0

Figura 19: Captura de pacotes TCP para tentativa de conexão SSH

Como pode ser visto na Figura 18, pacote 15, capturado pelo Wireshark, no pacote TCP, o segmento FIN estava sendo enviado, finalizando a conexão.

¹²Cisco Community, tokon, 2020.

¹³Network, 2020.

¹⁴StackExchange, UnixLinux, 2017.

¹⁵Cisco Community, valery.popov, 2009.

¹⁶Stack Exchange, serverfault, 2020.

¹⁷Consult, 2022.

```
gabriel@gabriel-VirtualBox:~$ ssh -l admin 192.168.20.3
Unable to negotiate with 192.168.20.3 port 22: no matching key exchange method found. Their offer: diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1
```

Figura 20: Finalização da tentativa de conexão da VM com R1

Além disso, foi tentado alterar configurações no firewall embutido em R1, permitindo a conexão de outros equipamentos à porta 22, utilizada pelo SSH, porém também sem sucesso.

5.4.3 A solução da conexão

Após diversas tentativas de estabelecimento de conexão entre a VM e R1, foi visto, na própria documentação do roteador Mikrotik¹⁸, que o serviço SSH é habilitado por padrão, sem que haja a necessidade de configurações adicionais para estabelecimento de conexões. Em seguida, foi tentado estabelecer a conexão entre a VM e o MK. Como o endereço IP de MK é obtido via DHCP, o mesmo foi consultado para que pudesse ser utilizado. No caso do endereço IP da VM, é configurado estaticamente. Após isso, foi verificado se MK estava ativo, por meio do ping, originário da VM, além de verificar qual rota os pacotes fariam, preferencialmente. Por fim, houve a tentativa de estabelecer a conexão SSH. Os pacotes trafegados estavam sendo capturados pelo Wireshark, por meio do enlace entre R1 e R2. A conexão SSH foi estabelecida com sucesso e os pacotes trafegados podem ser vistos no arquivo auxiliar de captura. Nesses pacotes, podem ser visto a troca de informações para início da conexão e outros dados, que retornam informações entre o cliente e o servidor.

5 6.170552 192.168.20.10	192.168.30.2	ICMP	98 Echo (ping) request id=0x0001, seq=1/256, ttl=63 (reply in 6)
6 0.055632 192.168.30.2	192.168.20.10	ICMP	98 Echo (ping) reply id=0x0001, seq=1/256, ttl=63 (request in 5)
7 0.919943 192.168.20.10	192.168.30.2	ICMP	98 Echo (ping) request id=0x0001, seq=2/512, ttl=63 (reply in 8)
8 0.012689 192.168.30.2	192.168.20.10	ICMP	98 Echo (ping) reply id=0x0001, seq=2/512, ttl=63 (request in 7)

Figura 21: Captura de pacotes via Wireshark, no ping de VM para MK

26 0.002930 192.168.1.2	192.168.20.10	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
27 0.000000 192.168.1.2	192.168.20.10	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
28 0.000000 192.168.1.2	192.168.20.10	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
29 0.006239 192.168.20.10	192.168.30.2	UDP	74 48684 → 33450 Len=32
30 0.000000 192.168.20.10	192.168.30.2	UDP	74 58079 → 33451 Len=32
31 0.000000 192.168.20.10	192.168.30.2	UDP	74 58279 → 33452 Len=32
32 0.003908 192.168.30.2	192.168.20.10	ICMP	102 Destination unreachable (Port unreachable)
33 0.000000 192.168.30.2	192.168.20.10	ICMP	102 Destination unreachable (Port unreachable)
34 0.000000 192.168.30.2	192.168.20.10	ICMP	102 Destination unreachable (Port unreachable)

Figura 22: Captura de pacotes via Wireshark, no traceroute de VM para MK

¹⁸Mikrotik, 2022.

53 4.009716 192.168.20.10	192.168.30.2	TCP	74 58758 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=3181010728 TSeqc=0 WS=128
54 0.014640 192.168.30.2	192.168.20.10	TCP	74 22 → 58758 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1460 SACK_PERM=1 TSeqval=4294960119 TSeqc=4294960119
55 0.017568 192.168.20.10	192.168.30.2	TCP	66 58758 + 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=3181010752 TSeqc=4294960119
56 0.000000 192.168.20.10	192.168.30.2	SSHv2	107 Client: Protocol (SSH-2.0-OpenSSH_8.2p1_Ubuntu-4ubuntu0.5)
57 0.025375 192.168.30.2	192.168.20.10	TCP	66 22 → 58758 [ACK] Seq=1 Ack=4 Win=14496 Len=0 TSeqval=4294960122 TSeqc=181010752
58 0.012440 192.168.30.2	192.168.20.10	SSHv2	82 Server: Protocol (SSH-2.0-ROSSH)
59 0.017568 192.168.30.2	192.168.20.10	TCP	66 22 → 58758 [ACK] Seq=1 Ack=42 Win=64256 Len=0 TSeqval=4294960122 TSeqc=4294960122
60 0.000000 192.168.20.10	192.168.30.2	TCP	1514 58758 → 22 [ACK] Seq=42 Ack=17 Win=64256 Len=1448 TSeqval=3181011593 TSeqc=4294960202 [TCP segment of a reassembled PDU]
61 0.000000 192.168.20.10	192.168.30.2	SSHv2	130 Client: Key Exchange Init
62 0.036112 192.168.30.2	192.168.20.10	TCP	66 22 → 58758 [ACK] Seq=17 Ack=1490 Win=17376 Len=0 TSeqval=4294960207 TSeqc=3181011593
63 0.000000 192.168.30.2	192.168.20.10	TCP	66 22 → 58758 [ACK] Seq=17 Ack=1554 Win=17376 Len=0 TSeqval=4294960208 TSeqc=3181011593
64 0.140544 192.168.30.2	192.168.20.10	SSHv2	514 Server: Key Exchange Init

Figura 23: Trecho da captura de pacotes via Wireshark, para conexão SSH entre VM e MK

```

gabriel@gabriel-VirtualBox:~$ ping 192.168.30.2
  G 192.168.30.2 (192.168.30.2) 56(84) bytes of data.
  4 bytes from 192.168.30.2: icmp_seq=1 ttl=62 time=94.3 ms

2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 26.832/60.565/94.299/33.733 ms
gabriel@gabriel-VirtualBox:~$ traceroute 192.168.30.2
traceroute to 192.168.30.2 (192.168.30.2), 30 hops max, 60 byte packets
  1 _gateway (192.168.20.3)  11.827 ms  11.986 ms  11.964 ms
  2  192.168.1.2 (192.168.1.2)  21.719 ms  21.975 ms  21.987 ms
  3  192.168.30.2 (192.168.30.2)  31.907 ms  32.072 ms  32.046 ms
gabriel@gabriel-VirtualBox:~$ ssh -l admin 192.168.30.2

MikroTik RouterOS 6.48.6 (c) 1999-2021      http://www.mikrotik.com/
[?]          Gives the list of available commands
command [?]  Gives help on the command and list of arguments
[Tab]         Completes the command/word. If the input is ambiguous,
              a second [Tab] gives possible options
/             Move up to base level
..            Move up one level
/command      Use command at the base level

[admin@RouterOS] > quit
interrupted
Connection to 192.168.30.2 closed.
gabriel@gabriel-VirtualBox:~$ 

```

Figura 24: Conexão estabelecida entre VM e MK

6 Conclusão

O projeto proposto visa buscar entender como os protocolos de roteamento e serviços das camadas de rede funcionam na prática. Além disso, foi possível trabalhar com diversas imagens de equipamentos que também são utilizados em topologias de ISPs. Diante disso, pode-se concluir que as configurações dos equipamentos, por mais que tenham havido algumas intercorrências, aconteceram de forma correta, bem como a utilização e configuração dos

protocolos de roteamento, serviços da camada de rede e aplicação. Além disso, foi possível entender o funcionamento de uma topologia de rede, bem como utilizar ferramentas para fazer a verificação de serviços de rede e a captura de pacotes trafegados.

7 Anexos

Como anexos a esse projeto, há alguns arquivos de configuração dos equipamentos. Para os equipamentos Cisco, a configuração foi obtida a partir da seguinte instrução:

```
show running-config
```

Para o caso do equipamento Mikrotik, a instrução executada foi:

```
export
```

Os arquivos com as capturas de pacotes, via Wireshark, configurações dos equipamentos e capturas de telas estão disponíveis via GitHub¹⁹.

¹⁹Bernardi, 2022.

Referências

- Auvik, Kevin Dooley (2014). *Configuring SSH on a Cisco Device*. <https://www.auvik.com/franklyit/blog/configuring-ssh-cisco-device/>. Acessado em: 2022-08-01.
- Bernardi G. R. (2022). *ArqRedesTCP IP Projeto*. https://github.com/gabrielrbernardi/ArqRedesTCP_IP_Projeto. Acessado em: 2022-08-02.
- Cisco (2015). *Secure Shell Configuration Guide, Cisco IOS Release 15S*. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_ssh/configuration/15-s/sec-usr-ssh-15-s-book.pdf. Acessado em: 2022-07-21.
- Cisco Community, tokon (2020). *C2960L-SM - ssh connection not possible*. <https://community.cisco.com/t5/switching/c2960l-sm-ssh-connection-not-possible/td-p/4257894>. Acessado em: 2022-08-01.
- Cisco Community, valery.popov (2009). *SSH access to c7200*. <https://community.cisco.com/t5/switching/ssh-access-to-c7200/td-p/1300786>. Acessado em: 2022-07-28.
- Cisco, sshamim, rvigil (2013). *OSPF Design Guide*. <https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html>. Acessado em: 2022-07-31.
- Cloud Flare (2022). *What is an autonomous system?* <https://www.cloudflare.com/learning/network-layer/what-is-an-autonomous-system/>. Acessado em: 2022-07-31.
- Consult M. (2022). *Configuring SSH on Cisco IOS*. <https://www.youtube.com/watch?v=kpuUOx6NZ0>. Acessado em: 2022-07-21.
- Fortinet (2022). *What Is Border Gateway Protocol (BGP)?* <https://www.fortinet.com/resources/cyberglossary/bgp-border-gateway-protocol>. Acessado em: 2022-07-31.
- IETF RFC 4251 (2022). *The Secure Shell (SSH) Protocol Architecture*. <https://datatracker.ietf.org/doc/html/rfc4251>. Acessado em: 2022-08-01.
- Microsoft (2022). *Dynamic Host Configuration Protocol (DHCP)*. <https://docs.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-top>. Acessado em: 2022-08-01.
- Mikrotik (2022). *Management tools, SSH*. <https://help.mikrotik.com/docs/display/ROS/SSH>. Acessado em: 2022-08-01.
- Must Be Geek, Arranda Saputra (2019). *Redistribute OSPF Route into BGP in Cisco IOS Router*. <https://www.mustbegeek.com/redistribute-ospf-route-into-bgp-in-cisco-ios-router/>. Acessado em: 2022-07-21.
- Network T. C. L. (2020). *SSH/Telnet from outside to self zone with Zone-base firewall*. <https://learningnetwork.cisco.com/s/question/0D53i00000KstykCAB/sshtelnet-from-outside-to-self-zone-with-zonebase-firewall>. Acessado em: 2022-07-28.

Solutions S. T. (2020). *How to Configure SSH on Cisco Router in GNS3*. <https://www.sysnettechsolutions.com/en/configure-ssh-gns3/>. Acessado em: 2022-07-21.

Stack Exchange, serverfault (2020). *What causes SSH error: kex_exchange_identification : Connectionclosedbyremotehost?*. <https://serverfault.com/questions/1015547/what-causes-ssh-error-kex-exchange-identification-connection-closed-by-remote>. Acessado em: 2022-07-30.

StackExchange, UnixLinux (2017). *How to enable diffie-hellman-group1-sha1 key exchange on Debian 8.0?* <https://unix.stackexchange.com/questions/340844/how-to-enable-diffie-hellman-group1-sha1-key-exchange-on-debian-8-0>. Acessado em: 2022-08-01.

Wireshark (2022). *About Wireshark*. <https://www.wireshark.org/>. Acessado em: 2022-08-01.