UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

LUIZ FERNANDO DE OLIVEIRA AMARAL

ANÁLISE COMPARATIVA DE DESEMPENHO DO PROTOCOLO DE ROTEAMENTO BORDER GATEWAY PROTOCOL

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2017

LUIZ FERNANDO DE OLIVEIRA AMARAL

ANÁLISE COMPARATIVA DE DESEMPENHO DO PROTOCOLO DE ROTEAMENTO BORDER GATEWAY PROTOCOL

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Bacharel em Sistemas de Informação" – Área de Concentração: Redes de Computadores.

Orientador: Fabiano Scriptore de Carvalho

CURITIBA

2017

Ministério da Educação
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Câmpus Curitiba
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Coordenação do Curso de Bacharelado em Sistemas de Informação

TERMO DE APROVAÇÃO

"ANÁLISE COMPARATIVA DE DESEMPENHO DO PROTOCOLO DE ROTEAMENTO BORDER GATEWAY PROTOCOL"

por

"Luiz Fernando de Oliveira Amaral"

Este Trabalho de Conclusão de Curso foi apresentado às **15:30 hs** do dia **29 de junho de 2017** como requisito parcial à obtenção do grau de Bacharel em Sistemas de Informação na Universidade Tecnológica Federal do Paraná - UTFPR - Câmpus Curitiba. O(a)(s) aluno(a)(s) foi(ram) arguido(a)(s) pelos membros da Banca de Avaliação abaixo assinados. Após deliberação a Banca de Avaliação considerou o trabalho _____.

_____ **Prof. Fabiano Scriptore de Carvalho** (Presidente - UTFPR/Curitiba)	_____ **Profa. Leyza Baldo Dorini** (Avaliador 1 - UTFPR/Curitiba)
_____ **Profa. Anelise Munaretto Fonseca** (Avaliador 2 - UTFPR/Curitiba)	_____ **Profa. Leyza Baldo Dorini** (Professor Responsável pelo TCC – UTFPR/Curitiba)
_____ **Prof. Leonelo Dell Anhol Almeida** (Coordenador(a) do curso de Bacharelado em Sistemas de Informação – UTFPR/Curitiba)	

"A Folha de Aprovação assinada encontra-se na Coordenação do Curso."

AGRADECIMENTOS

Agradeço a todas as pessoas que de alguma forma colaboraram não somente durante a elaboração deste Trabalho de Conclusão de Curso, mas também durante toda a minha jornada pela universidade.

Agradeço em especial aos meus familiares que sempre estiveram ao meu lado nos momentos difíceis, ao Prof. Fabiano pela amizade, pelo total apoio neste trabalho e também por despertar meu interesse pela área de Redes de Computadores, as Profas. Leyza e Mari por fazerem as disciplinas de programação do início do curso parecerem fáceis e a Profa. Marilia pela alegria nas aulas e pela dedicação e apoio durante a migração da grade do curso.

RESUMO

De Oliveira Amaral, Luiz Fernando. ANÁLISE COMPARATIVA DE DESEMPENHO DO PROTOCOLO DE ROTEAMENTO BORDER GATEWAY PROTOCOL. 70 f. Trabalho de Conclusão de Curso – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

O protocolo *Border Gateway Protocol*, mais conhecido como BGP, é um protocolo de roteamento externo cuja função é conectar os diversos sistemas autônomos da Internet através da troca de informações de roteamento e alcançabilidade. Atualmente encontra-se na sua quarta versão (BGP-4) e tem sido desde 1994 o principal protocolo de roteamento na Internet, sendo necessário para que os diversos *Internet Service Providers* (ISPs) possam estabelecer conexões uns aos outros. Portanto, dada a importância do BGP, este trabalho busca realizar uma análise comparativa de desempenho entre duas diferentes configurações do protocolo: uma configuração básica com os mínimos requisitos para se realizar o roteamento entre sistemas autônomos e uma segunda configuração com a implementação de autenticação TCP MD5, cujo objetivo é mitigar alguns tipos de ataque ao BGP. Estes dois cenários passaram por três testes padronizados que verificaram o tempo de resposta entre dois dispositivos finais em sistemas autônomos diferentes, o tempo de convergência do BGP para um enlace redundante e a utilização de CPU no roteador. Os resultados mostram que a opção do BGP com autenticação é vantajosa visto que o desempenho é levemente impactada.

Palavras-chave: BGP, Border Gateway Protocol, desempenho, TCP, MD5, sistemas autônomos, roteamento, Internet

ABSTRACT

De Oliveira Amaral, Luiz Fernando. BORDER GATEWAY PROTOCOL PERFORMANCE COMPARATIVE ANALYSIS. 70 f. Trabalho de Conclusão de Curso – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

The Border Gateway Protocol, also known as BGP, is an external routing protocol whose function is to connect the various autonomous systems on the Internet through the exchange of routing and reachability information. It is currently in the fourth version (BGP-4) and has been, since 19964, the main Internet routing protocol for Internet Service Providers (ISPs) to establish connections to each other. Therefore, given the importance of BGP, this paper aims to perform a comparative performance analysis between two different scenarios: a basic configuration with the minimum requirements to perform the routing between autonomous systems and another configuration with the MD5 TCP authentication implementation, whose objective is to mitigate some types of attack to the BGP. These two scenarios underwent three standardized tests that verified the response time between two end devices in different autonomous systems, the BGP convergence time to a redundant link and the CPU utilization in the router. The results showed that the option of BGP with authentication is preferable given that the performance was slightly impacted.

Keywords: BGP, Border Gateway Protocol, performance, TCP, MD5, autonomous systems, routing, Internet

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A Internet surgiu a partir da evolução da rede ARPANET, que foi desenvolvida pela *Advanced Research Projects Agency* (ARPA) do Departamento de Defesa dos Estados Unidos da América. Ela foi a primeira rede operacional comutada por pacotes, abrangendo inicialmente quatro localizações. Hoje em dia, a Internet atende bilhões de usuários e o número de conexões cresce exponencialmente (STALLINGS, 2007).

Para controlar tal expansão e proporcionar uma melhor infraestrutura para a Internet, protocolos de roteamento externo foram criados, separando domínios de roteamento entre diferentes entidades administrativas, mais conhecidas como sistemas autônomos.

No começo da Internet, o protocolo de roteamento utilizado era o *Exterior Gateway Protocol* (EGP). Porém, sua restrição de topologia e a ineficiência em lidar com loops de roteamento e o estabelecimento de políticas de roteamento abriram espaço para um protocolo mais robusto.

Deste então, o protocolo *Border Gateway Protocol* (BGP) é o protocolo utilizado para realizar o roteamento na Internet, mais especificamente o roteamento entre os diferentes sistemas autônomos. O BGP é capaz de lidar com centenas de milhares de rotas e, por isso, se faz crucial a sua compreensão para o correto funcionamento e operação da Internet (HALABI; MCPHERSON, 2000).

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

O presente trabalho tem como objetivo realizar uma análise comparativa de desempenho do protocolo de roteamento *Border Gateway Protocol* sendo utilizado no roteamento entre sistemas autônomos, que é conhecido como *Exterior Border Gateway Protocol* (eBGP). A análise se dará num ambiente de testes controlado. O BGP intradomínios, ou

Interior Border Gateway Protocol (iBGP), não faz parte do escopo deste trabalho.

1.1.2 OBJETIVOS ESPECÍFICOS

Como objetivos específicos, tem-se:

- Descrever a função dos sistemas autônomos;
- Identificar e descrever as características e funções do BGP;
- Analisar o comportamento do BGP numa rede de testes sob diferentes configurações;
- Analisar o impacto no desempenho do roteamento através da comparação de uma configuração básica do protocolo contra uma configuração acrescida do mecanismo de autenticação TCP MD5.

1.2 MOTIVAÇÃO E JUSTIFICATIVA

O funcionamento da Internet, considerando-se a nível global, depende fortemente do protocolo BGP. Este protocolo é essencial para se estabelecer o roteamento entre os diversos sistemas autônomos presentes no mundo todo. Caso ocorram falhas com as informações de roteamento compartilhadas entre tais sistemas autônomos, as suas respectivas redes e até mesmo as redes de outros sistemas autônomos podem ficar indisponíveis e afetar usuários e empresas que dependam dela.

O BGP é um dos protocolos mais complexos e difíceis de se configurar, tendo foco em segurança e escalabilidade. Por conta disso, grande parte dos problemas enfrentados por administradores de sistemas autônomos são causados por erros de configuração do protocolo (MOURA, 2001).

Além disso, a área de pesquisa relacionada à segurança do protocolo BGP é relativamente nova. Existem três fatores principais que acarretam em algumas vulnerabilidades do protocolo (MURPHY, 2006).

Assim, é de extrema importância que um administrador de redes e/ou sistemas autônomos venha a conhecer os tópicos que serão abordados neste trabalho e os adaptem para o seu contexto.

1.3 ESTRUTURA DO TRABALHO

Este primeiro capítulo apresentou o tema do trabalho e seus objetivos, além de mostrar a importância do BGP para o funcionamento da Internet. Em seguida, o segundo capítulo expõe o levantamento bibliográfico a respeito dos sistemas autônomos, protocolos de roteamento interno e o BGP em si. O terceiro capítulo apresenta o desenvolvimento deste trabalho e os recursos de *hardware* e *software* utilizados. O quarto capítulo apresenta a análise dos resultados obtidos e, por fim, as considerações finais e as referências bibliográficas são apresentadas.

2 REFERENCIAL TEÓRICO

Para se compreender o funcionamento do protocolo BGP, faz-se necessário entender alguns conceitos que estão por trás do seu funcionamento, como por exemplo, os diferentes tipos de protocolos de roteamento e o conceito de sistemas autônomos. Esta seção explicita esses conceitos, além do BGP em si.

2.1 PROTOCOLOS DE ROTEAMENTO

O roteamento é a principal forma de entrega de pacotes de dados entre dispositivos em uma rede comutada por pacotes. O roteador recebe um pacote de dados, abre-o, analisa o endereço IP de destino, calcula o próximo salto e o encaminha. Este processo se repete até que tal pacote chegue ao seu destino final. No entanto, para que esse processo funcione corretamente, são necessários protocolos de roteamento e suas respectivas tabelas de roteamento.

Os protocolos de roteamento podem ser classificados em três abordagens diferentes de acordo com a maneira como utilizam as informações de roteamento. Eles podem ser dos seguintes tipos: protocolo do tipo vetor de distância, protocolo do tipo estado de enlace e protocolo do tipo vetor caminho (MOURA, 1997).

A classe de protocolos do tipo vetor de distância, também referenciada como algoritmos *Bellman-Ford*, anuncia rotas como sendo vetores de distância e direção. A distância é definida em termos de uma métrica que representa o número de saltos, enquanto que a direção representa o roteador do próximo salto para determinada rede de destino. Sendo assim, cada roteador adquire rotas provenientes da perspectiva de seus vizinhos e as propaga de acordo com a sua própria perspectiva. Esta classe de protocolos utiliza um algoritmo de roteamento que envia atualizações de rotas para todos os vizinhos através do envio de sua tabela de roteamento completa.

Diferentemente dos protocolos vetor de distância, que possuem informações apenas de roteadores vizinhos, os protocolos do tipo estado de enlace possuem o conhecimento

de toda a rede. Isso porque os roteadores tem informação sobre todos os outros que utilizam o mesmo protocolo. Cada roteador gera informações sobre si mesmo, sobre os enlaces diretamente conectados à ele e seus respectivos estados. Esta informação se propaga por todos os roteadores sem ser alterada. Desta forma, todos eles possuem a mesma informação em relação à rede e podem calcular as melhores rotas de maneira independente (DOYLE, 1998).

Já os protocolos do tipo vetor caminho são protocolos que dispensam o uso de métricas para a escolha das melhores rotas, como por exemplo o número de saltos nos protocolos vetor de distância e a largura de banda nos protocolos estado de enlace. Ao invés disso, os protocolos do tipo vetor caminho simplesmente fornecem informações sobre quais redes um determinado roteador pode alcançar e qual o caminho que deve ser utilizado, ou seja, uma sequência de sistemas autônomos. Esta abordagem difere da abordagem vetor de distância em dois aspectos: primeiro que protocolos do tipo vetor caminho não fornecem a distância ou o custo para alcançar determinada rede de destino, e segundo que cada bloco de informação de roteamento lista todos os sistemas autônomos visitados para alcançar tal rede de destino (STALLINGS, 2007).

2.2 SISTEMAS AUTÔNOMOS

Visto o conceito dos protocolos do tipo vetor caminho, é preciso entender o que são os sistemas autônomos e como é estruturado o roteamento na Internet.

A arquitetura de roteamento da Internet é estruturada em dois níveis. No primeiro nível, a Internet é dividida em domínios onde cada um possui seu próprio ambiente de roteamento. Esses domínios utilizam protocolos do tipo *Interior Gateway Protocol* (IGP), que mantêm um mapeamento completo do domínio indicando o melhor caminho entre dois dispositivos. Esta abordagem, entretanto, não suporta o tamanho e complexidade da Internet. Os protocolos de roteamento interno mais utilizados são o *Open Shortest Path First* (OSPF), *Intermediate System-to-Intermediate System* (IS-IS) e *Enhanced Interior Gateway Routing Protocol* (EIGRP).

Já o segundo nível na hierarquia de roteamento é composto pelo roteamento inter-domínio. Neste ambiente é descrito apenas como os domínios se conectam, evitando descrever os caminhos dentro de cada um deles. Aqui, um caminho para um determinado destino é descrito como uma sequência de domínios que devem ser visitados para alcançar tal destino. O protocolo utilizado hoje em dia para manter este ambiente inter-domínios

é o *Border Gateway Protocol* na sua quarta versão (BGP-4).

Cada domínio de roteamento é um domínio administrativo único, operado dentro de políticas de roteamento independentes de outros domínios. Tais domínios são unidades autônomas na arquitetura de roteamento da Internet, sendo conhecidos como sistemas autônomos. A organização *Internet Engineering Task Force* (IETF), na RFC 4271, descreve um sistema autônomo como um conjunto de roteadores sob uma única administração técnica, usando um protocolo de roteamento interno (IGP) e métricas comuns para rotear pacotes dentro do sistema autônomo e um protocolo de roteamento externo (EGP) para realizar o roteamento de pacotes para outros sistemas autônomos (HUSTON, 2006).

Um sistema autônomo é um conjunto de roteadores e outros equipamentos de rede que compartilham de uma mesma política de roteamento, definida pela autoridade que administra tal sistema autônomo. Geralmente se utiliza um único protocolo de roteamento interno, porém pode ser que haja um conjunto de um ou mais protocolos dependendo de como está estrutura o sistema autônomo. A figura 1 exemplifica a relação entre os protocolos de roteamento interno, o BGP e os sistemas autônomos:

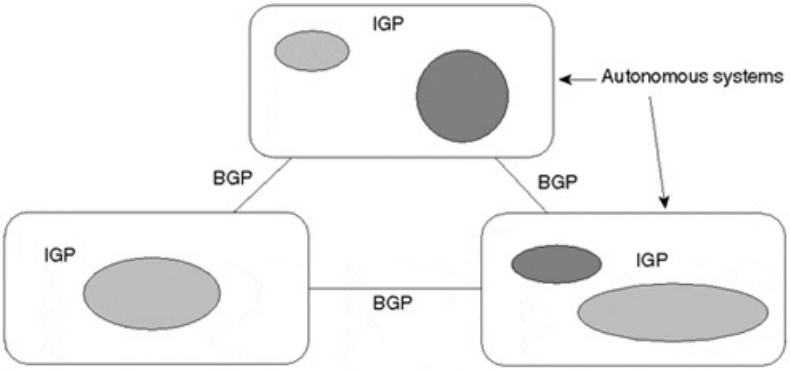


Figura 1: Relação entre sistemas autônomos.
Fonte: (HALABI; MCPHERSON, 2000)

A possibilidade de se ter uma enorme rede global, a Internet, só foi possível graças à sua segregação em diversas unidades administrativas menores e mais gerenciáveis. Seria impossível ter uma rede global operando com o protocolo OSPF ou o IS-IS, por exemplo, pois a quantidade de informações de roteamento seria incongruente. Tal segregação, representada pelos sistemas autônomos, possibilita a implementação de um conjunto de regras e políticas próprias que vão distinguir os diversos sistemas autônomos presentes no mundo todo. Cada um deles pode operar independentemente das regras e políticas dos

outros (HALABI; MCPHERSON, 2000).

2.3 PROTOCOLOS DE ROTEAMENTO INTERNO

Esta seção descreve alguns dos principais protocolos de roteamento interno utilizados atualmente.

2.3.1 ROUTING INFORMATION PROTOCOL

Routing Information Protocol, ou RIP, é um protocolo relativamente antigo do tipo IGP (do inglês*Interior Gateway Protocol*) e destinado a redes pequenas e homogêneas. Foi originalmente definido na RFC 1058 e, em 1994, foi atualizado pela RFC 1723, que descreve a sua segunda versão, RIPv2.

Protocolo do tipo *distance-vector*, o RIP envia atualizações de rota tanto em intervalos regulares como imediatamente após mudanças na rede. Quando um roteador recebe uma atualização, ele atualiza a sua tabela de roteamento e mantém apenas a melhor rota. Após esse processo, o roteador, através do conceito de *broadcast*, transmite a atualização para os seus vizinhos realizarem o mesmo processo (CISCO, 2012).

O tempo de convergência no RIP é relativamente alto devido aos seus *timers*. O envio de atualizações de roteamento é feito a cada 30 segundos. Se um roteador não receber uma atualização de um roteador vizinho dentro de um período de 180 segundos, ele marca as rotas provenientes de tal roteador como inutilizáveis. Caso esse tempo se estenda para além de 240 segundos, o roteador remove todas as entradas da tabela de roteamento referentes ao roteador vizinho que está sem comunicação.

A métrica utilizada pelo RIP é unicamente o número de saltos até a rede de destino. Entende-se por salto um roteador que esteja situado na rota. Uma rede diretamente conectada ao roteador tem métrica zero, enquanto que uma rede inalcançável tem métrica 16, ou seja, uma rede configurada com o protocolo RIP comporta distâncias de até 15 saltos (CISCO, 2006).

2.3.2 ENHANCED INTERIOR GATEWAY ROUTING PROTOCOL

Enhanced Interior Gateway Routing Protocol, ou EIGRP, é um protocolo IGP que fornece um tempo de convergência rápido e que utiliza o *Diffused Update Algorithm* (DUAL) para calcular o melhor caminho até uma rede de destino. Este protocolo é do tipo

vetor de distância com algumas características avançadas de protocolos do tipo estado de enlace (CISCO, 2005a).

Um roteador configurado com EIGRP armazena rotas alternativas, quando disponíveis, para as redes de destino. Desta forma, a convergência em caso de falha é rápida. Quando não há nenhuma rota alternativa disponível na tabela de roteamento, o EIGRP propaga uma consulta aos seus vizinhos para descobrir uma rota apropriada.

O uso reduzido da largura de banda é outra característica importante do EIGRP. As atualizações da tabela de roteamento são parciais, ou seja, quando as rotas mudam, o DUAL envia informações referentes apenas às mudanças e não a tabela de roteamento completa. Além disso, ao contrário do protocolo estado de enlace, que envia atualizações à todos os roteadores de uma área (*broadcast*), o EIGRP envia apenas aos roteadores que necessitam de tais atualizações (*multicast*) (PAQUET; TEARE, 2003).

Em relação à métrica de roteamento, que é utilizada para a decisão do melhor caminho até uma rede de destino, o EIGRP utiliza por padrão as métricas de largura de banda e de *delay* para o cálculo da métrica total (CISCO, 2015).

2.3.3 OPEN SHORTEST PATH FIRST

Open Shortest Path First, ou OSPF, é um protocolo estado de enlace do tipo IGP que foi desenvolvido pela *Internet Engineering Task Force* (IETF) e definido na *Request For Comments* (RFC) 2328. Se encaixa em redes TCP/IP e suporta *Variable Length Subnet Mask* (VLSM). Além disso, suporta autenticação em atualizações de roteamento, que são feitas utilizando-se IP *multicast*, sumarização de rotas e agregação de interfaces para se obter maiores larguras de banda (IETF, 1998).

Entre algumas características do OSPF, destaca-se a velocidade de convergência. Ela é bastante rápida devido às alterações de roteamento serem calculadas em paralelo e difundidas imediatamente, ao invés de se esperar por um tempo de *holddown* e envelhecimento de rota. Além disso, o uso da largura de banda é baixo pois atualizações de estado de enlace são enviadas apenas quando ocorrem mudanças na rede ou a cada 30 minutos (PAQUET; TEARE, 2003).

A métrica adotada pelo OSPF, também chamada de custo da interface, é a sobrecarga para se enviar pacotes de dados através desta interface. Tal custo é inversamente proporcional à largura de banda da interface, ou seja, altas larguras de banda indicam um baixo custo e, consequentemente, a melhor rota para encaminhar pacotes. Por padrão, o

custo é calculado de acordo com a largura de banda, mas pode ser manualmente ajustado com o comando `ip ospf cost 'valor'` (CISCO, 2005b).

2.4 BORDER GATEWAY PROTOCOL

O BGP se encaixa em uma categoria diferente dos protocolos descritos anteriormente. Tais protocolos operam dentro um único sistema autônomo e a sua principal função é realizar o roteamento de pacotes da forma mais eficiente possível. Por outro lado, o objetivo de um protocolo de roteamento interdomínios é diferente. No roteamento entre os diversos sistemas autônomos, os protocolos de roteamento devem se preocupar não só com eficiência, mas também com política. Tais políticas envolvem questões como segurança, relações econômicas e relações comerciais. Por exemplo, um sistema autônomo corporativo decide que poderá enviar e receber pacotes de qualquer outro sistema autônomo da Internet, mas não permitirá pacotes com origem ou destino que não sejam o seu próprio sistema autônomo de transitar através da sua rede. Ou então pode decidir que permitirá este tipo de tráfego mas que cobrará por tais serviços, como fazem as operadoras de Internet, por exemplo. Assim, percebe-se que há diversas variações de políticas de trânsito de dados (TANENBAUM; WETHERALL, 2011).

O BGP é um protocolo que possibilita a troca de informações de roteamento entre os diversos sistemas autônomos que compõem a Internet. Em contraste, os protocolos de roteamento interno (IGP) realizam o roteamento de informações somente dentro de um único sistema autônomo. A principal função do BGP é informar a alcançabilidade de outros sistemas BGP, formando assim um grafo de conectividade de sistemas autônomos através do qual loops de roteamento podem ser evitados (REKHTER et al., 2006).

A primeira versão do protocolo, a versão BGP-1, foi lançada em 1989 e passou por diversas mudanças até alcançar a versão BGP-4, em 1993, sendo introduzido nesta versão o suporte à endereços agregados (*Classless Interdomain Routing*, ou simplesmente CIDR) e o conceito de *supernets*. Isso foi possível graças ao suporte à prefixos IP para a divulgação de redes de destino e a eliminação do conceito de classes de endereços IP. Além disso, a versão 4 introduziu o conceito de agregação de rotas.

Na sua operação, o BGP assume que o roteamento dentro de um sistema autônomo é realizado por algum protocolo do tipo intradomínio, que pode ser por exemplo o RIP, EIGRP, OSPF e até mesmo rotas estáticas. Além disso, o BGP não impõe restrições na topologia de rede subjacente. Do ponto de vista do BGP, a Internet é um grafo bastante

complexo onde cada sistema autônomo, representado por um nó, possui um número único que o identifica. As conexões entre sistemas autônomos formam um caminho, e o conjunto de caminhos formam uma rota composta pelos números dos sistemas autônomos que devem ser percorridos para alcançar determinado destino (HALABI; MCPHERSON, 2000).

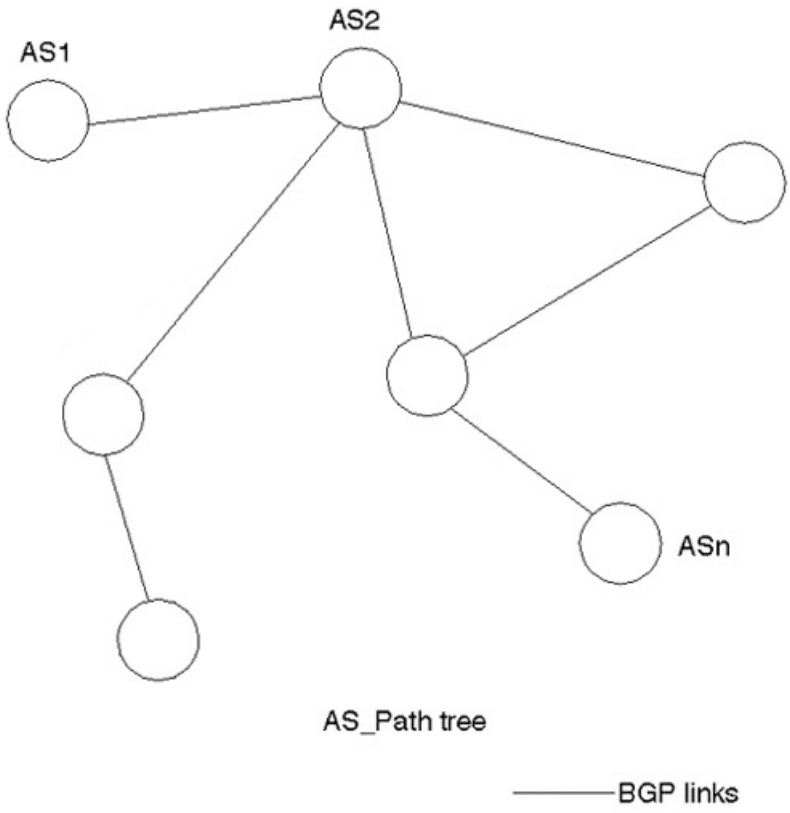


Figura 2: Exemplo de caminhos BGP.
Fonte: (HALABI; MCPHERSON, 2000)

O BGP utiliza o protocolo TCP na porta 179 para trocar informações de roteamento com outros sistemas BGP. Desta forma, o BGP se isenta do controle de transmissão das informações de roteamento (MOURA, 1999).

2.4.1 TIPOS DE MENSAGEM

O protocolo BGP opera em termos de mensagens que são enviadas por meio de uma conexão TCP. A negociação entre vizinhos BGP depende do correto processamento de mensagens *OPEN* e da detecção periódica de mensagens *UPDATE* ou *KEEPALIVE* (HALABI; MCPHERSON, 2000). O cabeçalho das mensagens BGP é composto por um

campo Marcador de 16 bytes, seguido de um campo Tamanho de 2 bytes e de um campo Tipo de 1 byte. A figura 3 ilustra o formato básico deste cabeçalho:

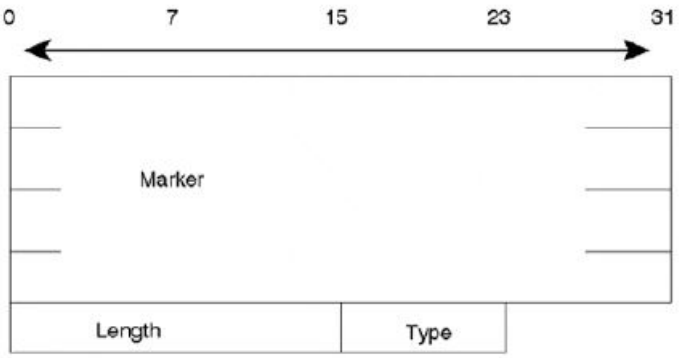


Figura 3: Formato do cabeçalho de uma mensagem BGP.
Fonte: (HALABI; MCPHERSON, 2000)

A lista a seguir traz as mensagens utilizadas pelo BGP e suas respectivas funções:

- *OPEN*: são mensagens utilizadas para iniciar uma sessão BGP com outro roteador;
- *UPDATE*: a função deste tipo de mensagem é transmitir informações sobre uma única rota e/ou listar diversas rotas que devem ser removidas;
- *KEEPALIVE*: mensagem utilizada para confirmar o recebimento de uma mensagem *OPEN* e também para periodicamente confirmar a relação de vizinhança BGP. É composta apenas pelo cabeçalho BGP, que é comum a todos os tipos de mensagens, seguido de nenhum dado adicional.
- *NOTIFICATION*: mensagem enviada quando algum erro é detectado;

Em relação ao formato das mensagens, o que todas têm em comum são os campos Marcador (*Marker*), Comprimento (*Length*) e Tipo (*Type*). O Marcador é um campo reservado para autenticação, onde o remetente pode inserir um valor de autenticação que será utilizado pelo destinatário como parte do mecanismo de autenticação. Desta forma, o destinatário é capaz de verificar a identidade do remetente. O campo Comprimento indica o comprimento total da mensagem em bytes. Já o campo Tipo especifica o tipo da mensagem. A figura 4 ilustra o formato de todas as mensagens BGP:

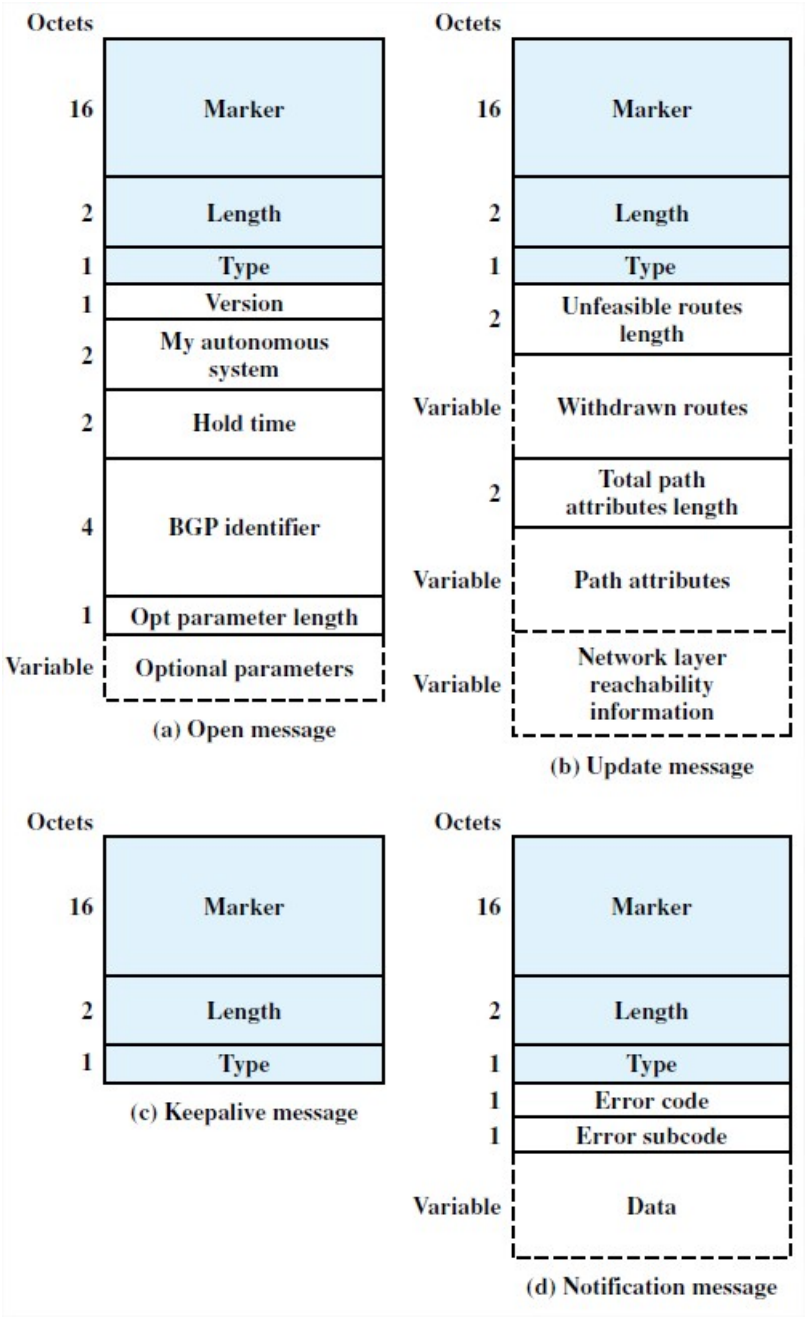


Figura 4: Formato das mensagens BGP.
Fonte: (STALLINGS, 2007)

2.4.2 ESTABELECENDO UMA SESSÃO BGP

O estabelecimento de uma sessão entre dois vizinhos BGP engloba diferentes estágios de negociação até que a sessão esteja completamente estabelecida. Se não for completada com sucesso, a troca de informações de atualização de rota não poderá ocorrer. A figura 5 ilustra os eventos deste processo de negociação:

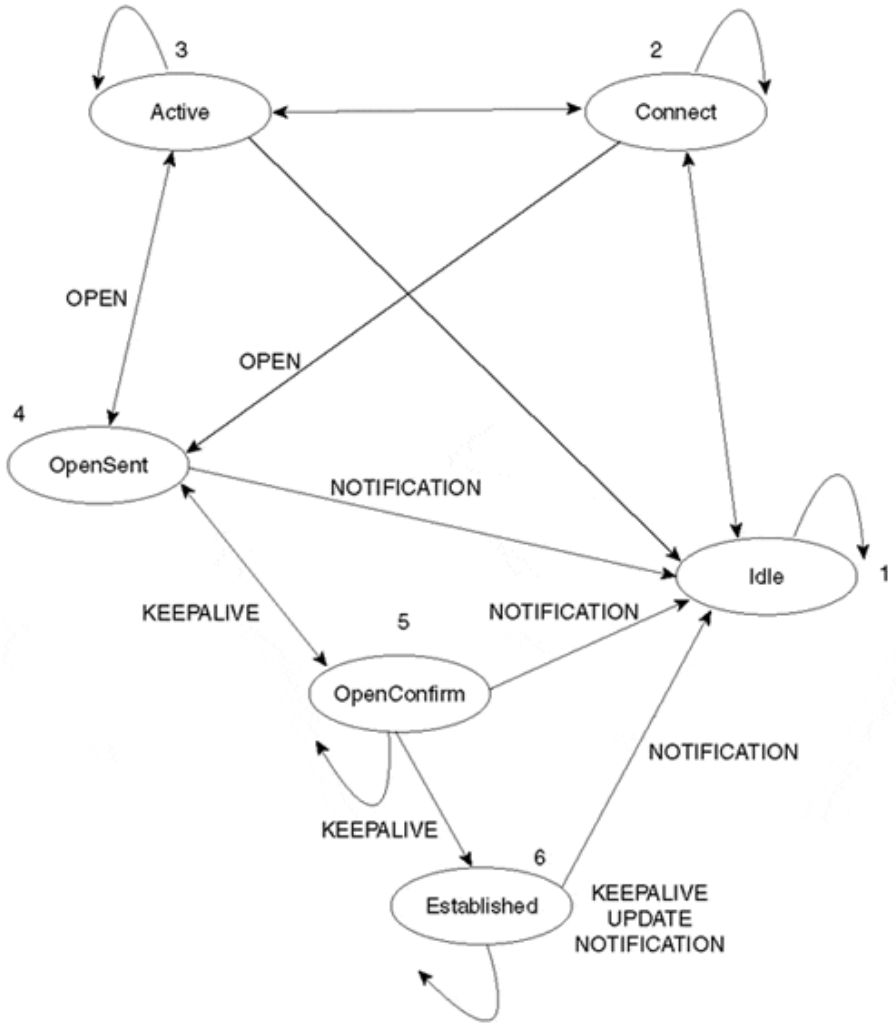


Figura 5: Estados de negociação entre vizinhos BGP.
Fonte: (HALABI; MCPHERSON, 2000)

A lista a seguir descreve cada uma das etapas no estabelecimento de uma sessão BGP.

- *Idle*: Neste primeiro estado, o BGP aguarda algum evento de inicialização que pode acontecer após um administrador de rede configurar o BGP num roteador ou após a reinicialização do BGP, por exemplo. Assim que o evento de inicialização acontece, o BGP inicializa seus recursos, reinicia seu timer *ConnectRetry*, inicia uma conexão TCP e passa a escutar por uma conexão que pode ser iniciada por algum outro sistema BGP remoto. Por fim, o BGP passa para o estado *Connect*. Em caso de falhas, ele retorna para o estado *Idle*;

- *Connect*: O BGP aguarda a conexão TCP se estabelecer e, caso seja bem sucedida, o estado muda para *OpenSent*. No caso de a conexão falhar, o estado muda para *Active* até o timer *ConnectRetry* expirar. Então, o contador é zerado e o estado muda para *Connect* e uma nova conexão é iniciada;

- *Active*: Neste estado o BGP tenta estabelecer uma conexão TCP com um *peer* e, em caso de sucesso, muda para o estado *OpenSent*. Caso contrário, o BGP retorna para o estado *Connect* e aguarda uma nova conexão ser iniciada por um *peer*. Em geral, a oscilação entre os estados *Connect* e *Active* indica algum problema com a conexão TCP;

- *OpenSent*: Aqui o BGP aguarda uma mensagem *OPEN* do seu *peer*. Se houve algum erro nesta mensagem, o sistema envia uma mensagem *NOTIFICATION* de erro e retorna ao estado *Idle*. Se não houver nenhum erro, o BGP começa e enviar mensagens *KEEPALIVE* e zera o *KEEPALIVE* timer. É no estado *OpenSent* que o BGP compara o seu número AS com o número do seu *peer* e identifica se eles pertecem ao mesmo sistema autônomo ou não;

- *OpenConfirm*: O BGP aguarda pela mensagem *KEEPALIVE* e, caso receba, o estado muda para *Established* e a negociação da sessão BGP está completa. Caso o BGP receba a mensagem *NOTIFICATION*, a sessão retorna para o estado *Idle*. O BGP envia mensagens *KEEPALIVE* periodicamente de acordo com o tempo definido no *KEEPALIVE* timer;

- *Established*: Este é o estado final da sessão BGP. O BGP passa a enviar mensagens *UPDATE* para seus vizinhos e, caso haja algum erro com esta mensagem, uma mensagem outra mensagem do tipo *NOTIFICATION* é enviada e o estado retorna para *Idle* (HALABI; MCPHERSON, 2000);

2.4.3 ATRIBUTOS DO BGP

Os atributos do BGP são um conjunto de parâmetros utilizados para controlar informações de rotas, como por exemplo caminho, grau de preferência de uma rota, o valor de *NEXT-HOP* e informações sobre agregação de rotas. Esses atributos são utilizados no processo de filtragem e escolha das melhoras rotas e podem ser divididos em quatro categorias:

- Conhecido Obrigatório (*Well-known mandatory*): atributos definidos na especi-

ficação do BGP que devem obrigatoriamente estar presentes nas mensagens de *UPDATE* e devem ser reconhecidos em todas as implementações do protocolo;

- Conhecido Arbitrário (*Well-Known Discretionary*): atributos também reconhecidos em todas as implementações do protocolo mas que são opcionais nas mensagens de *UPDATE*;

Além dos atributos do tipo *well-known*, existem outras duas categorias de atributos opcionais, que podem não estar presentes em todas as implementações do BGP:

- Opcional Transitivo (*Optional Transitive*): se uma determinada implementação do BGP recebe uma mensagem *UPDATE* com um atributo opcional não reconhecido, verifica-se a *flag transitive* e, caso esteja ativada, tal atributo é repassado nas mensagens *UPDATE* seguintes;

- Opcional Não Transitivo (*Optional Non-transitive*): ao contrário do anterior, caso a *flag transitive* não esteja ativada, o atributo não será repassado nas mensagens *UPDATE* (HALABI; MCPHERSON, 2000);

2.4.4 PROPAGAÇÃO DE ROTAS

Como já visto anteriormente, o BGP é uma espécie de protocolo do tipo vetor de distância. Entretanto, diferentemente dos protocolos intradomínio, ao invés de escolher rotas baseando-se no caminho mínimo, o BGP se baseia em políticas para realizar suas escolhas. Outra grande diferença é que o BGP não só mantém o custo de uma determinada rota, mas também o caminho de sistemas autônomos para se alcançar determinado destino. Devido à este fato, a abordagem deste protocolo é mais conhecida como vetor de caminho.

O caminho consiste no roteador do próximo salto, o que não significa que seja um roteador diretamente conectado ao roteador de origem, e na sequência de sistemas autônomos que a rota deve seguir. Pares de roteadores BGP se comunicam através de conexões TCP, desta forma a troca de informações é feita de maneira confiável.

A figura 6 exemplifica a propagação de rotas entre domínios BGP:

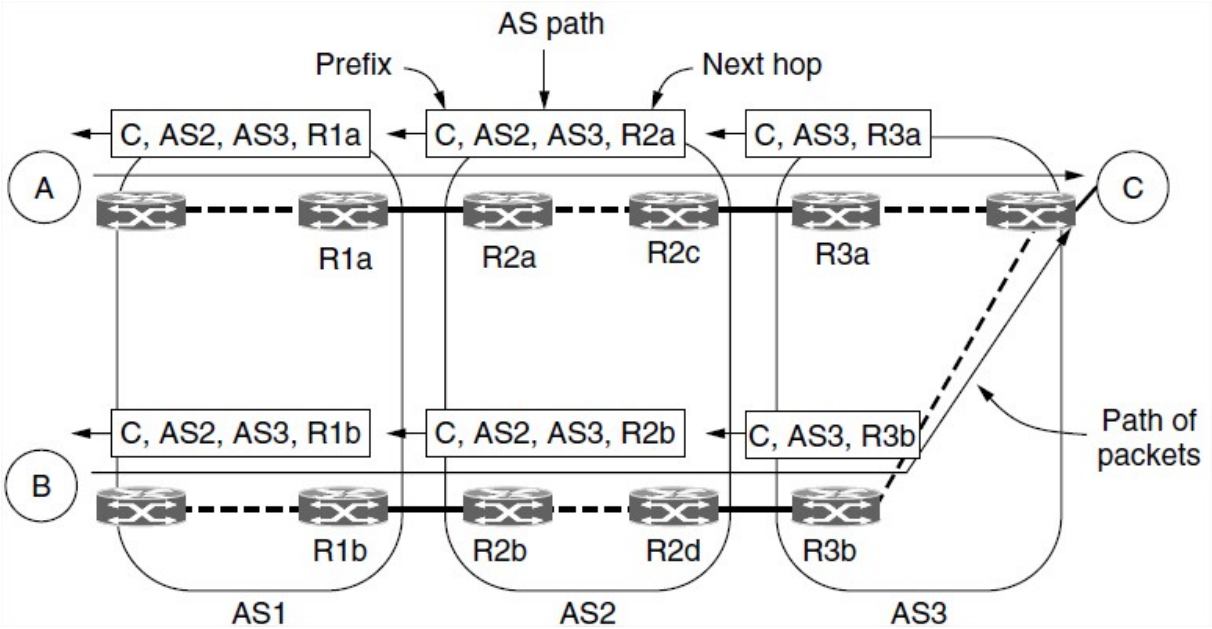


Figura 6: Propagação de rotas BGP.
Fonte: (TANENBAUM; WETHERALL, 2011)

Neste exemplo, a rede C, que se encontra no sistema autônomo AS3, está sendo anunciada para os demais sistemas autônomos, AS1 e AS2. Inicialmente, quando a rota está sendo propagada ao roteador R2c, ela contém apenas o caminho composto pelo sistema autônomo AS3 o roteador do próximo salto R3a. No enlace inferior, o caminho de sistemas autônomos é o mesmo (apenas o AS3), mas o roteador do próximo salto muda já que a rota foi propagada por um enlace diferente. Assim, se o roteador R2c, por exemplo, quiser enviar um pacote de dados para a rede C, ele sabe que deverá encaminhar o pacote para o roteador do próximo salto, o R3a, e que o caminho até a rede C passará apenas pelo AS3.

A propagação de rota continua e chega ao sistema autônomo AS1. No roteador R1a, o caminho de sistemas autônomos é AS2, AS3 e o roteador do próximo salto é o R2a. Desta forma, ao se manter o caminho completo de sistemas autônomos, o roteador que está recebendo uma atualização de rota consegue identificar *loops* de roteamento. Todo roteador que envia uma rota para fora do sistema autônomo insere o seu número na rota. Ao receber a rota, o roteador verifica se o seu próprio sistema autônomo já faz parte do caminho e, em caso afirmativo, o *loop* é detectado e a atualização de rota é descartada.

Após entendido como é feita a propagação de rotas no BGP, podemos agora descrever como os roteadores BGP escolhem qual rota utilizar para um determinado

destino. Cada roteador BGP aprende uma rota através de um roteador presente em outro próximo sistema autônomo ao qual ele está conectado, além dos outros roteadores de borda do mesmo sistema autônomo que estão conectados a outros roteadores em outros sistemas autônomos. Então, cada roteador deve decidir qual a melhor rota entre todas essas existentes e isso é feito de acordo com as políticas estabelecidas pela autoridade administrativa do sistema autônomo, que são ISPs ou outras empresas, por exemplo. Porém, observa-se que há algumas estratégias comuns aos sistemas autônomos.

A primeira delas é utilizar rotas através de sistemas autônomos parceiros, onde há uma relação de troca de tráfego ao invés de rotas através de provedores de tráfego, pois estas possuem um custo financeiro.

A segunda estratégia é escolher o menor caminho, ou seja, aquele com o menor número de sistemas autônomos. Ainda assim, não se pode ter certeza se o menor caminho é o mais eficiente, uma vez que a topologia e as políticas de roteamento utilizadas por cada um dos sistemas autônomos são desconhecidos. Entretanto, o caminho mais curto tende a ser o melhor em média, por isso essa estratégia é comumente utilizada.

Por fim, a terceira estratégia consiste em escolher a rota de menor custo dentro do próprio sistema autônomo. No exemplo da figura anterior, pacotes enviados da rede A para a rede C sairiam do AS1 através do roteador R1a, assim como pacotes da rede B sairiam através do roteador R1b. Ambos estão escolhendo o caminho de menor custo para sair do AS1. Essa estratégia é conhecida como *early exit* ou *hot-potato*. Portanto, percebe-se que roteadores BGP localizados em diferentes partes de um sistema autônomo podem escolher caminhos diferentes para alcançar uma mesma rede de destino (TANENBAUM; WETHERALL, 2011).

2.4.5 VULNERABILIDADES

Segurança em BGP ainda é uma área de pesquisa relativamente nova, onde a maioria das pesquisas buscam cobrir aspectos como integridade, autenticação, confidencialidade, autorização e validação de dados do protocolo BGP.

De acordo com a RFC 4272 (MURPHY, 2006), existem três fatores principais que causam vulnerabilidades no protocolo BGP. O primeiro deles é o fato de que o BGP não possui nenhum mecanismo interno para proporcionar uma forte proteção da integridade e autenticidade em comunicações *peer-to-peer*. Este fato motivou a utilização do algotirmo de hash TCPMD5, o que garante a integridade de mensagens BGP e a autenticação de

peers. Em segundo lugar, o BGP não é capaz de validar a autoridade de um sistema autônomo em divulgar informações de roteamento (prefixos de rede). Por último, não há nenhum mecanismo que assegure a autenticidade de atributos de rota propagados por sistemas autônomos.

Existem algumas variações do BGP que buscam cobrir tais vulnerabilidades do protocolo. O S-BGP, por exemplo, utiliza uma infraestrutura de chave pública para mitigar a falsificação de endereços IP. Através da utilização de certificados digitais, verifica-se quais endereços IP e quais números de sistemas autônomos estão realmente alocados aos sistemas BGP. Já o SO-BGP verifica a validade de um prefixo IP que está sendo divulgado. Ele verifica se o parceiro BGP que está divulgando determinado prefixo possui pelo menos um caminho válido até o destino. O PG-BGP mantém um histórico com as origens dos sistemas autônomos e prefixos de rede. Desta forma, novos sistemas autônomos e prefixos ficam num estado de menor preferência durante 24 horas. Passado esse período de monitoramento, caso eles ainda estejam na base de informações de roteamento, então eles são adicionados ao histórico do PG-BGP (WANG et al., 2009).

2.4.6 AUTENTICAÇÃO DE SESSÕES BGP

O protocolo BGP pode ser configurado para autenticar suas sessões com outros parceiros BGP. Ele suporta o mecanismo de autenticação baseado em *hash* chamado *Message Digest 5* (MD5), sendo recomendada a sua utilização pelo menos com os parceiros EBGP. Através da autenticação de sessões BGP, previne-se o roubo de uma sessão TCP, TCP *resets* além da injeção de rotas inconsistentes. Utilizando o MD5, o protocolo TCP cria um código *hash* para cada pacote que é enviado numa sessão BGP (JAIN; EDGEWORTH, 2016).

2.5 FUNÇÃO HASH MD5

O objetivo de uma função *hash* é comprovar a autenticidade de mensagens, arquivos, documentos ou simplesmente dados. A autenticação desses dados permite que agentes de comunicação verifiquem se mensagens recebidas são autênticas, ou seja, se o conteúdo delas não foi alterado e se o remetente é quem realmente deveria ser.

Uma função *hash* recebe de entrada uma mensagem M de tamanho variável e produz como saída um *message digest* ou *hash* $\mathrm{H}(M)$ com um tamanho fixo de 128 bits. Então, esse *hash* é enviado juntamente com a mensagem para que o destinatário possa

validar a autenticidade da mensagem.

Para que a autenticação de uma mensagem seja possível, as duas partes envolvidas na comunicação, A e B, precisam compartilhar uma chave secreta. Quando A envia uma mensagem para B, ele calcula a funçao *hash* sobre a concatenação da chave secreta com a mensagem. Como B também possui a chave secreta, ele consegue recalcular o *hash* da mensagem e comparar com o *hash* recebido. Como a chave secreta não é enviada e é conhecida apenas pelos parceiros da comunicação, um interceptador não consegue modificar mensagens capturadas, além de não conseguir gerar uma mensagem falsa.

Para que possa autenticar mensagens, uma função *hash* deve possuir os seguintes requerimentos:

- Os dados de entrada podem ser de qualquer tamanho.
- Os dados de saída possuem um tamaho fixo.
- A função *hash* deve ser relativamente fácil de computar.
- Dado um código *hash*, é computacionalmente inviável encontrar a sua mensagem de origem.
- É computacionalmente inviável gerar dois códigos *hash* idênticos a partir de dados diferentes.
- É computacionalmente inviável encontrar um par de mensagens que gerem o mesmo código *hash*.

É importante notar, através do quarto item da lista, que não se pode fazer o processo inverso da função, ou seja, a partir da mensagem e do seu *hash* descobrir a chave secreta utilizada, ou seja, é uma função de uma única direção. Essa propriedade é importante pois o MD5 utiliza uma chave secreta no processo de geração do *hash*, a qual não é enviada. Caso a função não fosse de uma única direção, interceptadores conseguiriam descobrir a chave secreta (STALLINGS, 2007).

2.6 SESSÕES BGP VIA OPÇÃO DE ASSINATURA TCP MD5

Existem algumas práticas de segurança que protegem o protocolo BGP contra ataques simples. É o caso da extensão TCP MD5. Essa extensão é, na verdade, uma nova opção do protocolo TCP que permite inserir um *hash* MD5, ou *Message Digest*, em

um segmento TCP. Esse *hash* se torna a assinatura do remetente neste segmento TCP e tal informação só faz sentido aos dois parceiros da comunicação BGP, uma vez que o MD5 utiliza uma chave compartilhada secreta entre ambos, conforme visto anteriormente. Assim, uma vez que o BGP utiliza o protocolo TCP na camada de transporte, esta nova opção reduz a probabilidade de ataques de roubo da sessão TCP, TCP *resets* e a injeção de rotas inconsistentes. Para conseguir falsificar um segmento TCP, um atacante precisaria descobrir não somente os números da sequência TCP, mas também a senha utilizada no algoritmo de *hash* MD5. Outra observação importante é que não há como parceiros BGP negociarem automaticamente a utilização da opção TCP MD5, uma vez que isso é uma questão de política de configuração que deve ser adotada por ambas as administrações dos sistemas autônomos envolvidos (HEFFERNAN, 1998).

Em abril de 2004, o Centro de Atendimento a Incidentes de Segurança (CAIS) da RNP divulgou um alerta no qual é descrito a vulnerabilidade que pode afetar a implementação do protocolo TCP. Esta vulnerabilidade pode permitir que um atacante consiga um estado de Negação de Serviço (DoS, na sigla em inglês) em conexões TCP que resultam no término de sessões. Por utilizar o protocolo TCP, o BGP foi considerado como a maior preocupação nessa situação, dado que ele é o protocolo de roteamento mais utilizado na Internet. Um ataque ao BGP causaria a queda de sessões BGP e a consequente perda das rotas previamente divulgadas (CAIS, 2004).

3 DESENVOLVIMENTO

Com base no levantamento bibliográfico realizado e nos conhecimentos adquiridos a respeito do protocolo de roteamento BGP, uma pesquisa de campo foi realiza a fim de se levantar as questões abordadas na seção de objetivos geral e específicos. A estratégia de pesquisa utilizada é a metodologia de estudo de caso, pois, segundo Yin (2013), esta é a abordagem que trata de acontecimentos em seu contexto real, que é exatamente o que foi buscado neste trabalho e nos testes planejados. O trabalho foi dividido nas seguintes partes:

- Pesquisa sobre BGP: nesta etapa inicial, o levantamento bibliográfico acerca do protocolo BGP foi realizado;
- Planejamento da infraestrutura: nesta etapa foram levantados e planejados os requisitos para a elaboração da infraestrutura de rede;
- Implementação da infraestrutura: assim que o planejamento foi concluído, a etapa seguinte abordou a implementação da infraestrutura de rede, abrangendo a organização física dos equipamentos de rede, desktops e cabeamento;
- Configuração dos equipamentos: após instalados, os equipamentos foram configurados com seus respectivos protocolos de roteamento;
- Fase de testes: nesta etapa foram realizados testes de desempenho do protocolo BGP em diferentes cenários de configuração. Tais testes serão detalhados na próxima seção;
- Coleta de dados: foram feitas coletas de dados após cada um dos testes realizados;
- Análise dos resultados: os resultados dos testes foram analisados e compilados.

O laboratório de redes da UTFPR, o LabRedes, e seus respectivos equipamentos de rede foram utilizados na elaboração de cenários de teste do protocolo BGP. Tais cenários

foram utilizados para testar o protocolo BGP em diferentes configurações e os resultados obtidos foram apresentados na forma de melhores práticas que podem ser aplicadas num contexto real de utilização do protocolo. A infraestrutura conta com roteadores Cisco 2801 e Cisco 1841, que serviram tanto como roteadores de borda quanto roteadores internos nos sistemas autônomos. Além disso, desktops foram utilizados como dispositivos finais para geração de tráfego na rede. A figura 7 ilustra a topologia que foi montada no laboratório de redes para a realização dos testes e análise do protocolo BGP:

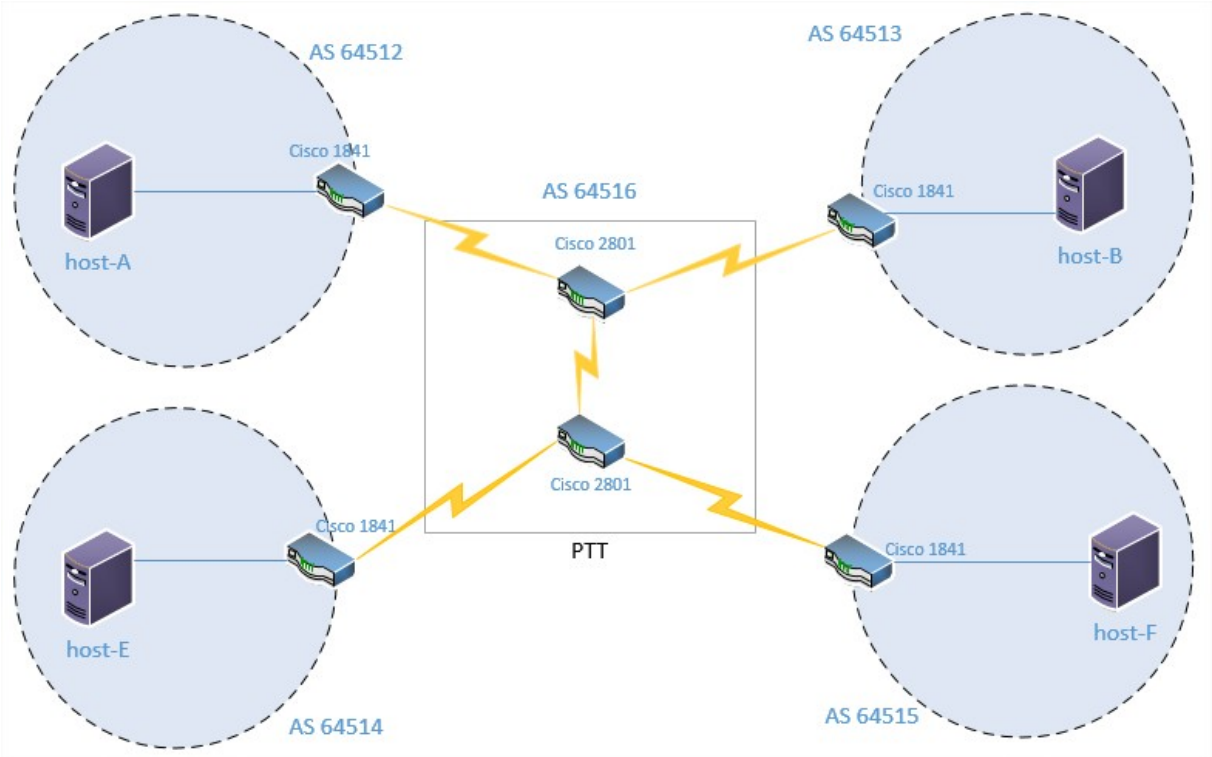


Figura 7: Topologia da rede.
Fonte: autoria própria.

3.1 RECURSOS DE HARDWARE

A base para o desenvolvimento deste trabalho foi o laboratório de redes da UTFPR (LabRedes), localizado no campus Curitiba, sede Centro. Foram utilizados seis roteadores no total. Destes, quatro foram utilizados como roteadores de borda e os dois restantes como parte do Ponto de Troca de Tráfego (PTT). As especificações dos roteadores encontram-se a seguir:

- Roteadores de borda:

Cisco 1800 Series	Cisco 1841
DRAM	Default: 256 MB
Memória Flash	Default: 64 MB
Slots Modulares	2
Portas Ethernet	2 de 10/100 MB
Porta Console	1 de até 115.2 kbps
Porta Auxiliar	1 de até 115.2 kbps

Tabela 1: Especificações do roteador Cisco 1841.
Fonte: autoria própria.

- Roteadores do PTT:

Cisco 2800 Series	Cisco 2801
DRAM	Default: 128 MB
Memória Flash	Default: 64 MB
Slots Modulares	4
Portas Ethernet	Duas de 10/100 MB
Porta Console	Uma de até 115.2 kbps
Porta Auxiliar	Uma de até 115.2 kbps

Tabela 2: Especificações do roteador Cisco 2801.
Fonte: autoria própria.

3.2 RECURSOS DE SOFTWARE

Com relação aos recursos de *software*, eles podem ser divididos em 3 categorias principais: sistema operacional dos roteadores, sistema operacional dos desktops e *software* para geração de tráfego.

O sistema operacional dos roteadores é fornecido pela Cisco e é chamado de Cisco *Internetwork Operating System*, ou Cisco IOS. A versão utilizada é a Cisco IOS Software Release 12.4, que é suporta pelos dois modelos de roteadores utilizados, Cisco 1841 e Cisco 2801.

Os desktops utilizam o sistema operacional Debian, o qual é gratuito e atende aos requisitos para a elaboração deste trabalho.

Por fim, foi utilizado o *software* JPerf na sua versão 2.0.2, uma ferramenta para testes de desempenho em redes de computadores. Ele suporta tanto o protocolo TCP como o UDP, além de redes IPv4 e IPv6. Seus testes incluem: medição de largura de banda, perda de pacotes, delay e jitter (DUGAN et al.,).

3.3 IMPLEMENTAÇÃO E TESTES

Nesta seção serão descritas as configurações aplicadas aos roteadores e o detalhamento dos testes realizados no LabRedes. Foram preparados dois cenários de testes com diferentes configurações do protocolo BGP em cada um deles. Em seguida, os mesmos testes foram realizados nos dois cenários. A figura 8 ilustra a topologia, as interfaces de rede e os endereços utilizados como base em ambos os cenários, alterando-se apenas a configuração do protocolo BGP.

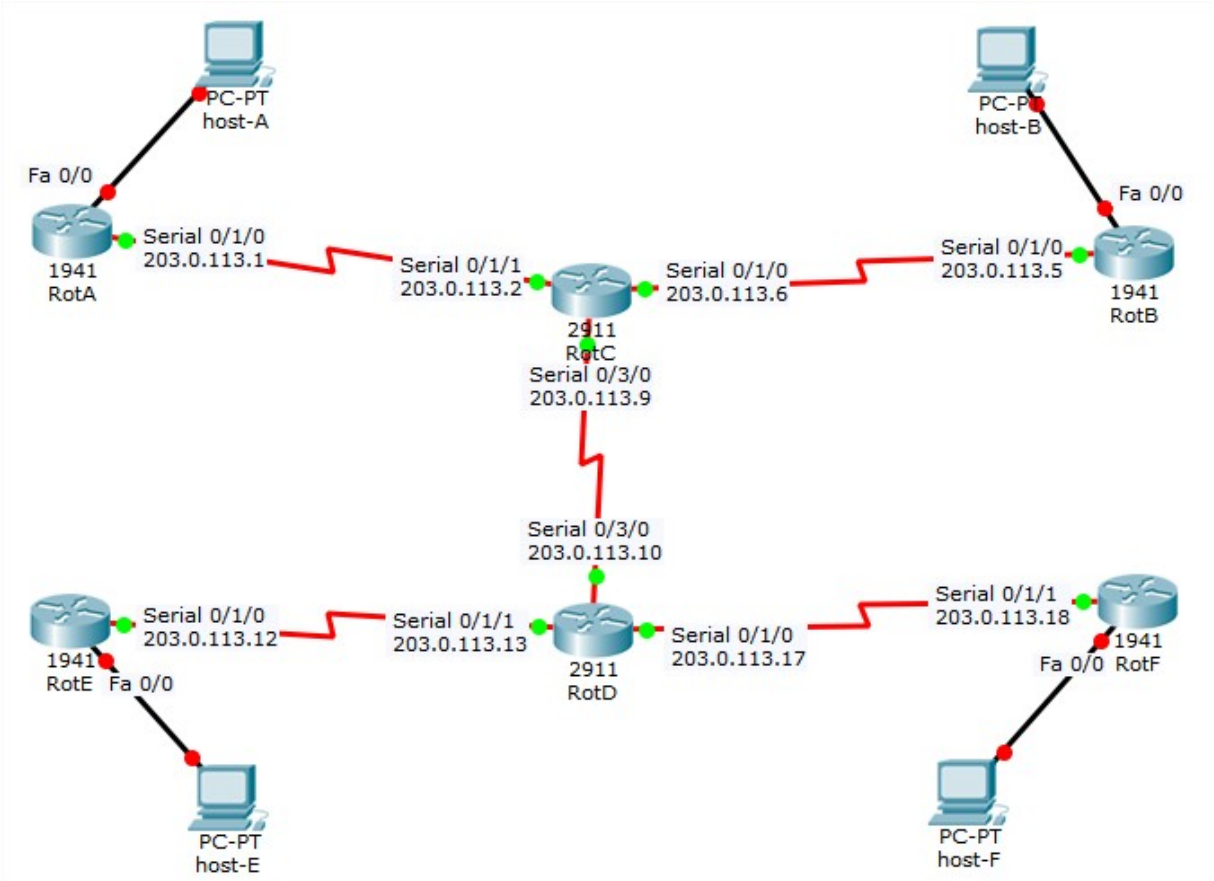


Figura 8: Interfaces e endereços IP.

Fonte: autoria própria.

3.3.1 CENÁRIO DE TESTE 1 - BGP BÁSICO

Neste primeiro cenário, todos os roteadores foram configurados para utilizar o protocolo BGP em seus respectivos sistemas autônomos. Os comandos na figura 9 mostram, como exemplo, as configurações BGP aplicadas ao roteador RotA:

```
RotA(config)# router bgp 64512
RotA(config-router)# neighbor 203.0.113.2 remote-as 64516
RotA(config-router)# network 10.1.1.0 mask 255.255.255.0
RotA(config-router)# network 203.0.113.0 mask 255.255.255.252
```

Figura 9: Exemplo de configuração do BGP.
Fonte: autoria própria.

O comando **router bgp** ***"número AS"*** inicializa o protocolo BGP no roteador e define seu número de sistema autônomo. Este número é único no roteador, ou seja, um roteador só pode pertencer a um único sistema autônomo. Já o comando **neighbor** ***"endereço IP"*** **remote-as** ***"número AS do parceiro"*** define o parceiro BGP com o qual informações de roteamento serão trocadas. Este comando é importante pois um roteador BGP não descobre um outro roteador BGP automaticamente. Ambos devem estar configurados para se comunicaram um com o outro. Por fim, o comando **network** ***"endereço de rede"*** **mask** ***"máscara de rede"*** define as redes diretamente conectadas ao roteador e que serão divulgadas aos demais parceiros BGP (CISCO, 2007).

Todos os demais roteadores foram configurados seguindo este padrão de comandos, estabelecendo sessões BGP com seus roteadores vizinhos.

3.3.1.1 TESTE 1: SOBRECARGA DE DADOS NA REDE

Este teste consiste em executar o comando **ping** para transmitir 20 pacotes de dados a partir do host-A com destino ao host-F em dois momentos diferentes. Num primeiro momento, executa-se o comando sem nenhum tráfego sendo gerado na rede. Desta forma, temos os dados de alcançabilidade e tempo de resposta numa situacão ótima. Em seguida, executa-se o comando **ping** novamente entre o host-A e host-F, só que dessa vez com uma carga de dados sendo gerada entre o host-B e o host-E através do *software* JPerf. A figura 10 mostra como foi configurado o *software* JPerf, tanto no host-B (cliente) como no host-E (servidor):

Figura 10: Cliente e servidor do software JPerf.

Fonte: autoria própria.

Para o comando **ping** em situação ótima (sem tráfego na rede), obteve-se o seguinte resultado:

```
PING 10.4.4.2 (10.4.4.2) 56(84) bytes of data.
64 bytes from 10.4.4.2: icmp_seq=1 ttl=60 time=2.78 ms
64 bytes from 10.4.4.2: icmp_seq=2 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=3 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=4 ttl=60 time=2.72 ms
64 bytes from 10.4.4.2: icmp_seq=5 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=6 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=7 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=8 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=9 ttl=60 time=2.72 ms
64 bytes from 10.4.4.2: icmp_seq=10 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=11 ttl=60 time=2.72 ms
64 bytes from 10.4.4.2: icmp_seq=12 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=13 ttl=60 time=2.71 ms
64 bytes from 10.4.4.2: icmp_seq=14 ttl=60 time=2.72 ms
64 bytes from 10.4.4.2: icmp_seq=15 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=16 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=17 ttl=60 time=2.72 ms
64 bytes from 10.4.4.2: icmp_seq=18 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=19 ttl=60 time=2.72 ms
64 bytes from 10.4.4.2: icmp_seq=20 ttl=60 time=2.74 ms

— 10.4.4.2 ping statistics —
20 packets transmitted, 20 received, 0% packet loss, time 19033ms
rtt min/avg/max/mdev = 2.717/2.736/2.788/0.027 ms
```

Figura 11: Ping do host-A ao host-F sem tráfego na rede.
Fonte: autoria própria.

A média do tempo de resposta dos 20 pacotes transmitidos foi de **2,736 ms** e nenhum pacote foi perdido.

Já o comando **ping** durante a geração de tráfego na rede obteve o seguinte resul-

tado:

```
  PING 10.4.4.2 (10.4.4.2) 56(84) bytes of data.
64 bytes from 10.4.4.2: icmp_seq=1 ttl=60 time=8.49 ms
64 bytes from 10.4.4.2: icmp_seq=2 ttl=60 time=9.01 ms
64 bytes from 10.4.4.2: icmp_seq=3 ttl=60 time=6.26 ms
64 bytes from 10.4.4.2: icmp_seq=4 ttl=60 time=12.5 ms
64 bytes from 10.4.4.2: icmp_seq=5 ttl=60 time=7.10 ms
64 bytes from 10.4.4.2: icmp_seq=6 ttl=60 time=11.2 ms
64 bytes from 10.4.4.2: icmp_seq=7 ttl=60 time=11.2 ms
64 bytes from 10.4.4.2: icmp_seq=8 ttl=60 time=8.22 ms
64 bytes from 10.4.4.2: icmp_seq=9 ttl=60 time=9.43 ms
64 bytes from 10.4.4.2: icmp_seq=10 ttl=60 time=6.24 ms
64 bytes from 10.4.4.2: icmp_seq=11 ttl=60 time=5.27 ms
64 bytes from 10.4.4.2: icmp_seq=12 ttl=60 time=7.65 ms
64 bytes from 10.4.4.2: icmp_seq=13 ttl=60 time=7.09 ms
64 bytes from 10.4.4.2: icmp_seq=14 ttl=60 time=11.0 ms
64 bytes from 10.4.4.2: icmp_seq=15 ttl=60 time=4.86 ms
64 bytes from 10.4.4.2: icmp_seq=16 ttl=60 time=12.1 ms
64 bytes from 10.4.4.2: icmp_seq=17 ttl=60 time=10.4 ms
64 bytes from 10.4.4.2: icmp_seq=18 ttl=60 time=11.6 ms
64 bytes from 10.4.4.2: icmp_seq=19 ttl=60 time=8.40 ms
64 bytes from 10.4.4.2: icmp_seq=20 ttl=60 time=7.02 ms

— 10.4.4.2 ping statistics —
20 packets transmitted, 20 received, 0% packet loss, time 19023ms
rtt min/avg/max/mdev = 4.864/8.773/12.554/2.298 ms
```

Figura 12: Ping do host-A ao host-F com geração de tráfego na rede. Fonte: autoria própria.

A média do tempo de resposta dos 20 pacotes transmitidos foi de **8,773 ms** e nenhum pacote foi perdido.

3.3.1.2 TESTE 2: TEMPO DE CONVERGÊNCIA

O objetivo deste teste é medir o tempo de convergência do protocolo BGP para um enlace redundante em caso de falha no enlace principal. Para isso, foi configurado um novo enlace entre o roteador RotA, na interface serial 0/0/0, e o roteador RotE, também na interface serial 0/0/0, conforme ilustrado na figura 13:

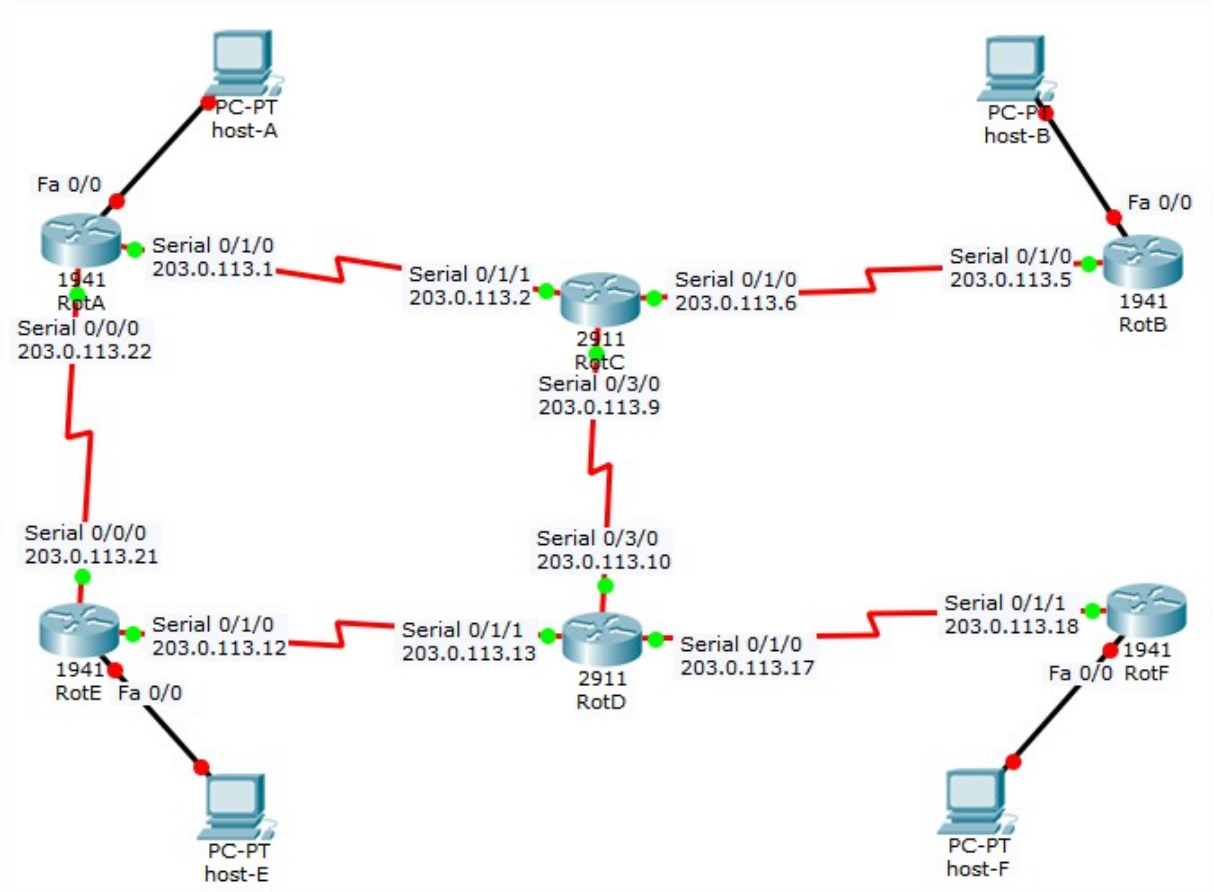


Figura 13: Topologia de rede com enlace redundante.
Fonte: autoria própria.

Para poder simular a falha de um determinado enlace, primeiramente foi executado o comando **traceroute** a partir do host-A com destino ao host-F para se verificar qual o caminho utilizado pelo BGP. O resultado do comando é mostrado na figura 14:

```
traceroute to 10.4.4.2 (10.4.4.2), 30 hops max, 60 byte packets
1 10.1.1.1 (10.1.1.1) 0.881 ms 1.069 ms 1.303 ms
2 203.0.113.2 (203.0.113.2) 1.372 ms 1.733 ms 1.970 ms
3 203.0.113.10 (203.0.113.10) 2.874 ms 3.162 ms 3.399 ms
4 203.0.113.18 (203.0.113.18) 4.093 ms 5.114 ms 5.736 ms
5 10.4.4.2 (10.4.4.2) 4.842 ms 5.464 ms 6.102 ms
```

Figura 14: Traceroute antes da falha de interface.
Fonte: autoria própria.

Percebe-se que o caminho utilizado foi através do roteador RotC, uma vez que o segundo salto aconteceu no endereço IP 203.0.113.2. Portanto, para verificar o tempo de convergência para o novo enlace, foi executado o comando **ping** para transmitir 50 pacotes de dados entre os dispositivos host-A e host-F seguido do desligamento forçado da interface serial 0/3/0 do RotC, o que é feito através do comando **shutdown** dentro das configurações de interface do roteador. Isso obriga o protocolo BGP a convergir para o novo enlace entre os roteadores RotA e RotE. Devido à quantidade de *pings*, o resultado do comando foi encurtado e é mostrado na figura 15:

PING 10.4.4.2 (10.4.4.2) 56(84) bytes of data. 64 bytes from 10.4.4.2: icmp_seq=1 ttl=60 time=2.78 ms
64 bytes from 10.4.4.2: icmp_seq=2 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=3 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=4 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=5 ttl=60 time=2.73 ms
...
64 bytes from 10.4.4.2: icmp_seq=10 ttl=60 time=2.72 ms
From 203.0.113.2 icmp_seq=11 Destination Host Unreachable
From 203.0.113.2 icmp_seq=13 Destination Host Unreachable
From 203.0.113.2 icmp_seq=15 Destination Host Unreachable
From 203.0.113.2 icmp_seq=16 Destination Host Unreachable
From 203.0.113.2 icmp_seq=17 Destination Host Unreachable
From 203.0.113.2 icmp_seq=18 Destination Host Unreachable
From 203.0.113.2 icmp_seq=19 Destination Host Unreachable
From 203.0.113.2 icmp_seq=20 Destination Host Unreachable
...
From 203.0.113.2 icmp_seq=36 Destination Host Unreachable
From 203.0.113.2 icmp_seq=37 Destination Host Unreachable
From 203.0.113.2 icmp_seq=38 Destination Host Unreachable
From 203.0.113.2 icmp_seq=39 Destination Host Unreachable
From 203.0.113.2 icmp_seq=40 Destination Host Unreachable
64 bytes from 10.4.4.2: icmp_seq=41 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=42 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=43 ttl=60 time=2.72 ms
64 bytes from 10.4.4.2: icmp_seq=44 ttl=60 time=2.74 ms
...
64 bytes from 10.4.4.2: icmp_seq=50 ttl=60 time=2.70 ms

— 10.4.4.2 ping statistics —
50 packets transmitted, 20 received, +28 errors, 60% packet loss,
time 49078ms
rtt min/avg/max/mdev = 2.704/2.734/2.782/0.039 ms

Figura 15: Falha da interface durante o comando ping.

Fonte: autoria própria.

Dos 50 pacotes transmitidos, apenas 20 foram recebidos, o que indica que o protocolo BGP, em sua configuração padrão neste cenário 1, perdeu 30 pacotes de dados durante a convergência para o novo enlace.

Após a execução do comando **ping**, o comando **traceroute** foi novamente executado a partir do host-A em direção ao host-F para verificar o novo caminho utilizado para a transmissão de dados:

```
traceroute to 10.4.4.2 (10.4.4.2), 30 hops max, 60 byte packets
1 10.1.1.1 (10.1.1.1) 1.185 ms 1.535 ms 2.247 ms
2 203.0.113.21 (203.0.113.21) 1.325 ms 1.708 ms 1.952 ms
3 203.0.113.13 (203.0.113.13) 2.916 ms 3.158 ms 5.522 ms
4 203.0.113.18 (203.0.113.18) 6.021 ms 5.754 ms 5.239 ms
5 10.4.4.2 (10.4.4.2) 4.975 ms 4.588 ms 4.170 ms
```

Figura 16: Traceroute após a falha de interface.
Fonte: autoria própria.

Percebe-se que o caminho utilizado após a falha da interface serial 0/3/0 do RotC foi através do RotE, já que o segundo salto é feito pelo endereço IP 203.0.113.21.

3.3.1.3 TESTE 3: UTILIZAÇÃO DE CPU

Para este teste, foi observada a utilização de CPU do roteador RotC sob uma sobrecarga de dados na rede obtida pela geração de tráfego de dados pelo *software* JPerf entre os dispositivos host-B e host-E, conforme já detalhado no Teste 1 e na Figura 10.

A utilização de CPU pode ser medida através do comando **show processes cpu**, conforme mostrado na figura 17:

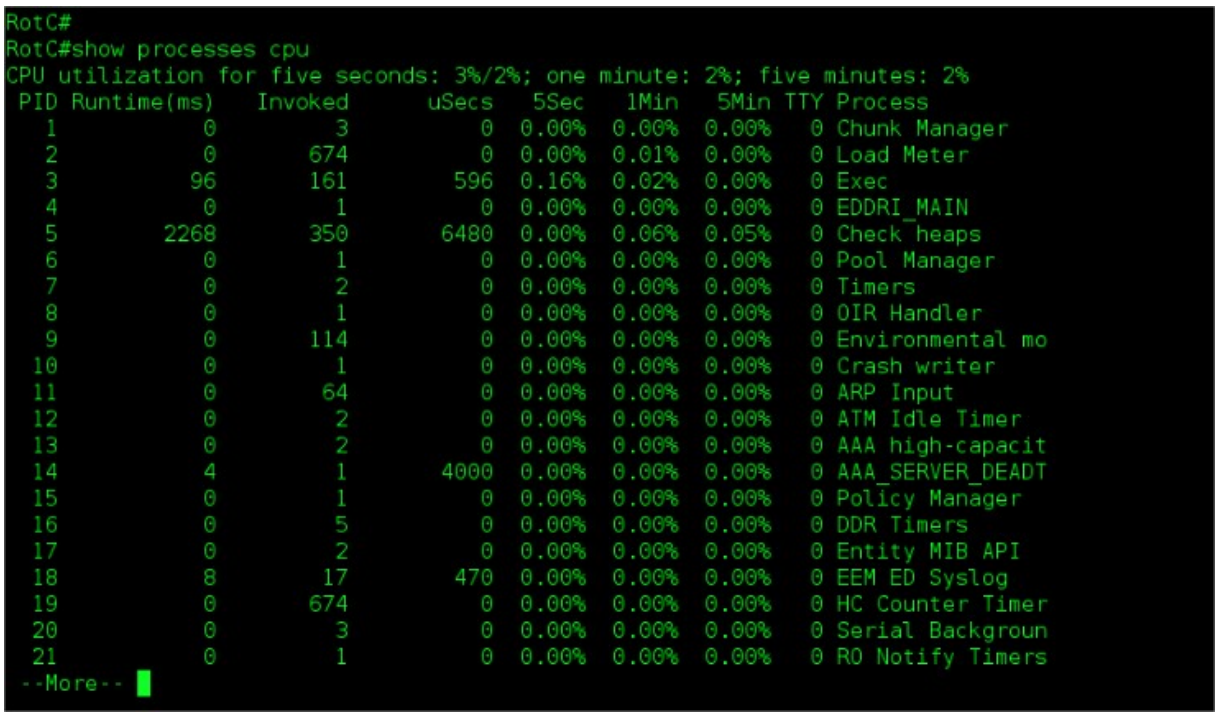


Figura 17: Utilização de CPU no RotC com a configuração básica do BGP.
Fonte: autoria própria.

A utilização de CPU atingiu uma média total de 3%.

3.3.2 CENÁRIO DE TESTE 2 - BGP COM AUTENTICAÇÃO TCP MD5

Este segundo cenário de teste consiste em todas as configurações já aplicadas no cenário de teste 1, ou seja, uma configuração básica do BGP, além da inclusão da autenticação via assinatura TCP MD5. Os comandos da figura 18 mostram, por exemplo, as configurações aplicadas ao roteador RotA:

```
RotA(config)# router bgp 64512
RotA(config-router)# neighbor 203.0.113.2 remote-as 64516
RotA(config-router)# network 10.1.1.0 mask 255.255.255.0
RotA(config-router)# network 203.0.113.0 mask 255.255.255.252
RotA(config-router)# neighbor 203.0.113.2 password tccbgppass
```

Figura 18: Exemplo de configuração do BGP com autenticação.
Fonte: autoria própria.

O comando **neighbor 203.0.113.2 password tccbgppass** habilita o mecanismo de autenticação através do algoritmo MD5. É importante lembrar que a autenticação deve ser habilitada nas duas pontas. Portanto, no roteador RotC deve ser executado o comando **neighbor 203.0.113.1 password tccbgppass**. Uma vez habilitada a autenticação, qualquer segmento TCP transmitido que pertença ao BGP é verificado e aceito somente se a autenticação obtiver sucesso. Para isso, ambos os roteadores devem utilizar a mesma chave secreta e, caso a autenticação falhe, a relação de vizinhança BGP não é estabelecida.

Todos os demais roteadores foram configurados de acordo com o padrão apresentado acima nas configurações do roteador RotA.

3.3.2.1 TESTE 1: SOBRECARGA DE DADOS NA REDE

Este teste foi realizado da mesma maneira como foi feito no cenário 1. Foram transmitidos 20 pacotes de dados através do comando **ping** a partir do host-A com destino ao host-F em dois momentos diferentes.

Para o comando **ping** em situação ótima (sem tráfego na rede), obteve-se o resultado mostrado na figura 19:

```
PING 10.4.4.2 (10.4.4.2) 56(84) bytes of data.
64 bytes from 10.4.4.2: icmp_seq=1 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=2 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=3 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=4 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=5 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=6 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=7 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=8 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=9 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=10 ttl=60 time=2.78 ms
64 bytes from 10.4.4.2: icmp_seq=11 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=12 ttl=60 time=2.77 ms
64 bytes from 10.4.4.2: icmp_seq=13 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=14 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=15 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=16 ttl=60 time=2.75 ms
```

```
64 bytes from 10.4.4.2: icmp_seq=17 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=18 ttl=60 time=2.77 ms
64 bytes from 10.4.4.2: icmp_seq=19 ttl=60 time=2.77 ms
64 bytes from 10.4.4.2: icmp_seq=20 ttl=60 time=2.76 ms

— 10.4.4.2 ping statistics —
20 packets transmitted, 20 received, 0% packet loss, time 19041ms
rtt min/avg/max/mdev = 2.742/2.763/2.780/0.031 ms
```

Figura 19: Ping do host-A com destino ao host-F.
Fonte: autoria própria.

A média do tempo de resposta dos 20 pacotes transmitidos foi de **2,763 ms** e nenhum pacote foi perdido.

Já o comando **ping** durante a geração de tráfego na rede obteve o resultado mostrado na figura 20:

```
PING 10.4.4.2 (10.4.4.2) 56(84) bytes of data.
64 bytes from 10.4.4.2: icmp_seq=1 ttl=60 time=11.0 ms
64 bytes from 10.4.4.2: icmp_seq=2 ttl=60 time=13.4 ms
64 bytes from 10.4.4.2: icmp_seq=3 ttl=60 time=8.16 ms
64 bytes from 10.4.4.2: icmp_seq=4 ttl=60 time=10.2 ms
64 bytes from 10.4.4.2: icmp_seq=5 ttl=60 time=11.0 ms
64 bytes from 10.4.4.2: icmp_seq=6 ttl=60 time=12.7 ms
64 bytes from 10.4.4.2: icmp_seq=7 ttl=60 time=8.01 ms
64 bytes from 10.4.4.2: icmp_seq=8 ttl=60 time=8.65 ms
64 bytes from 10.4.4.2: icmp_seq=9 ttl=60 time=10.5 ms
64 bytes from 10.4.4.2: icmp_seq=10 ttl=60 time=11.7 ms
64 bytes from 10.4.4.2: icmp_seq=11 ttl=60 time=12.9 ms
64 bytes from 10.4.4.2: icmp_seq=12 ttl=60 time=13.7 ms
64 bytes from 10.4.4.2: icmp_seq=13 ttl=60 time=7.02 ms
64 bytes from 10.4.4.2: icmp_seq=14 ttl=60 time=11.3 ms
64 bytes from 10.4.4.2: icmp_seq=15 ttl=60 time=10.7 ms
```

```
64 bytes from 10.4.4.2: icmp_seq=16 ttl=60 time=9.30 ms
64 bytes from 10.4.4.2: icmp_seq=17 ttl=60 time=9.60 ms
64 bytes from 10.4.4.2: icmp_seq=18 ttl=60 time=11.2 ms
64 bytes from 10.4.4.2: icmp_seq=19 ttl=60 time=11.1 ms
64 bytes from 10.4.4.2: icmp_seq=20 ttl=60 time=10.9 ms

— 10.4.4.2 ping statistics —
20 packets transmitted, 20 received, 0% packet loss, time 19027ms
rtt min/avg/max/mdev = 7.027/10.674/13.769/1.776 ms
```

Figura 20: Ping do host-A com destino ao host-F.
Fonte: autoria própria.

A média do tempo de resposta dos 20 pacotes transmitidos foi de **10,674 ms** e nenhum pacote foi perdido.

3.3.2.2 TESTE 2: TEMPO DE CONVERGÊNCIA

Assim como no cenário 1, o objetivo deste teste é medir o tempo de convergência do protocolo BGP para um enlace redundante em caso de falha no enlace principal. O enlace redundante utilizado foi o mesmo entre os roteadores RotA e RotE.

Para poder simular a falha de um determinado enlace, primeiramente foi executado o comando **traceroute** a partir do host-A com destino ao host-F para se verificar qual o caminho utilizado pelo BGP. O resultado do comando é mostrado na figura 21:

```
traceroute to 10.4.4.2 (10.4.4.2), 30 hops max, 60 byte packets
1 10.1.1.1 (10.1.1.1) 0.905 ms 1.124 ms 1.324 ms
2 203.0.113.2 (203.0.113.2) 1.377 ms 1.728 ms 1.965 ms
3 203.0.113.10 (203.0.113.10) 2.908 ms 3.182 ms 3.427 ms
4 203.0.113.18 (203.0.113.18) 4.101 ms 5.102 ms 5.727 ms
5 10.4.4.2 (10.4.4.2) 4.833 ms 5.452 ms 6.070 ms
```

Figura 21: Traceroute antes da falha de interface.
Fonte: autoria própria.

Percebe-se que o caminho utilizado, assim como aconteceu no cenário 1, foi através do roteador RotC, uma vez que o segundo salto aconteceu no endereço IP 203.0.113.2. Portanto, para verificar o tempo de convergência para o novo enlace, foi executado o comando **ping** para transmitir 50 pacotes de dados entre os dispositivos host-A e host-F seguido do desligamento forçado da interface serial 0/3/0 do RotC. Isso obriga o protocolo BGP a convergir para o novo enlace entre os roteadores RotA e RotE. O resultado do comando **ping** é mostrado na figura 22:

```
PING 10.4.4.2 (10.4.4.2) 56(84) bytes of data.
64 bytes from 10.4.4.2: icmp_seq=1 ttl=60 time=2.77 ms
64 bytes from 10.4.4.2: icmp_seq=2 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=3 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=4 ttl=60 time=2.77 ms
64 bytes from 10.4.4.2: icmp_seq=5 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=6 ttl=60 time=2.77 ms
64 bytes from 10.4.4.2: icmp_seq=7 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=8 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=9 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=10 ttl=60 time=2.75 ms
From 203.0.113.2 icmp_seq=11 Destination Host Unreachable
From 203.0.113.2 icmp_seq=13 Destination Host Unreachable
From 203.0.113.2 icmp_seq=15 Destination Host Unreachable
From 203.0.113.2 icmp_seq=16 Destination Host Unreachable
From 203.0.113.2 icmp_seq=17 Destination Host Unreachable
From 203.0.113.2 icmp_seq=18 Destination Host Unreachable
From 203.0.113.2 icmp_seq=19 Destination Host Unreachable
From 203.0.113.2 icmp_seq=20 Destination Host Unreachable
From 203.0.113.2 icmp_seq=21 Destination Host Unreachable
From 203.0.113.2 icmp_seq=22 Destination Host Unreachable
From 203.0.113.2 icmp_seq=23 Destination Host Unreachable
From 203.0.113.2 icmp_seq=24 Destination Host Unreachable
From 203.0.113.2 icmp_seq=25 Destination Host Unreachable
From 203.0.113.2 icmp_seq=26 Destination Host Unreachable
From 203.0.113.2 icmp_seq=27 Destination Host Unreachable
From 203.0.113.2 icmp_seq=28 Destination Host Unreachable
```

```
From 203.0.113.2 icmp_seq=29 Destination Host Unreachable
From 203.0.113.2 icmp_seq=30 Destination Host Unreachable
From 203.0.113.2 icmp_seq=31 Destination Host Unreachable
From 203.0.113.2 icmp_seq=32 Destination Host Unreachable
From 203.0.113.2 icmp_seq=33 Destination Host Unreachable
From 203.0.113.2 icmp_seq=34 Destination Host Unreachable
From 203.0.113.2 icmp_seq=35 Destination Host Unreachable
From 203.0.113.2 icmp_seq=36 Destination Host Unreachable
From 203.0.113.2 icmp_seq=37 Destination Host Unreachable
From 203.0.113.2 icmp_seq=38 Destination Host Unreachable
From 203.0.113.2 icmp_seq=39 Destination Host Unreachable
From 203.0.113.2 icmp_seq=40 Destination Host Unreachable
64 bytes from 10.4.4.2: icmp_seq=41 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=42 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=43 ttl=60 time=2.73 ms
64 bytes from 10.4.4.2: icmp_seq=44 ttl=60 time=2.76 ms
64 bytes from 10.4.4.2: icmp_seq=45 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=46 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=47 ttl=60 time=2.75 ms
64 bytes from 10.4.4.2: icmp_seq=48 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=49 ttl=60 time=2.74 ms
64 bytes from 10.4.4.2: icmp_seq=50 ttl=60 time=2.75 ms

— 10.4.4.2 ping statistics —
50 packets transmitted, 20 received, +28 errors, 60time 49087ms
rtt min/avg/max/mdev = 2.732/2.753/2.779/0.069 ms
```

Figura 22: Falha da interface durante o comando ping.
Fonte: autoria própria.

Dos 50 pacotes transmitidos, apenas 20 foram recebidos, o que indica que o protocolo BGP, em sua configuração com autenticação TCP MD5 do cenário 2, perdeu 30 pacotes de dados durante a convergência para o novo enlace.

Após a execução do comando **ping**, o comando **traceroute** foi novamente exe-

cutado a partir do host-A em direção ao host-F para verificar o novo caminho utilizado para a transmissão de dados:

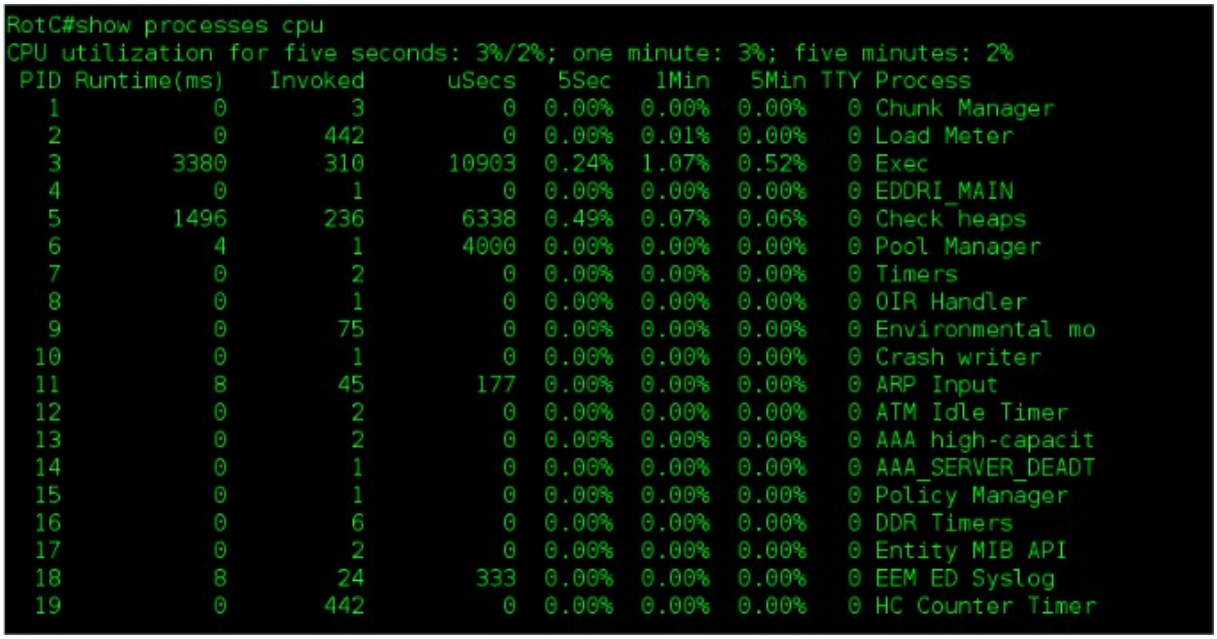
```
traceroute to 10.4.4.2 (10.4.4.2), 30 hops max, 60 byte packets
1 10.1.1.1 (10.1.1.1) 1.223 ms 1.569 ms 2.276 ms
2 203.0.113.21 (203.0.113.21) 1.377 ms 1.903 ms 1.978 ms
3 203.0.113.13 (203.0.113.13) 2.978 ms 3.210 ms 5.528 ms
4 203.0.113.18 (203.0.113.18) 6.021 ms 5.758 ms 5.245 ms
5 10.4.4.2 (10.4.4.2) 4.984 ms 4.584 ms 4.195 ms
```

Figura 23: Traceroute após a falha de interface.
Fonte: autoria própria.

Percebe-se que o caminho utilizado após a falha da interface serial 0/3/0 do RotC foi, assim como no cenário 1, através do RotE, já que o segundo salto é feito pelo endereço IP 203.0.113.21.

3.3.2.3 TESTE 3: UTILIZAÇÃO DE CPU

Para este teste, foi seguido o mesmo procedimento aplicado no teste de utilização de CPU do cenário 1. O resultado pode ser visto na figura 24:



```
RotC#show processes cpu
CPU utilization for five seconds: 3%/2%; one minute: 3%; five minutes: 2%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
   1           0         3          0  0.00%  0.00%  0.00%   0 Chunk Manager
   2           0       442          0  0.00%  0.01%  0.00%   0 Load Meter
   3        3380       310      10903  0.24%  1.07%  0.52%   0 Exec
   4           0         1          0  0.00%  0.00%  0.00%   0 EDDRI_MAIN
   5        1496       236       6338  0.49%  0.07%  0.06%   0 Check heaps
   6           4         1       4000  0.00%  0.00%  0.00%   0 Pool Manager
   7           0         2          0  0.00%  0.00%  0.00%   0 Timers
   8           0         1          0  0.00%  0.00%  0.00%   0 OIR Handler
   9           0        75          0  0.00%  0.00%  0.00%   0 Environmental mo
  10           0         1          0  0.00%  0.00%  0.00%   0 Crash writer
  11           8        45        177  0.00%  0.00%  0.00%   0 ARP Input
  12           0         2          0  0.00%  0.00%  0.00%   0 ATM Idle Timer
  13           0         2          0  0.00%  0.00%  0.00%   0 AAA high-capacit
  14           0         1          0  0.00%  0.00%  0.00%   0 AAA_SERVER_DEADT
  15           0         1          0  0.00%  0.00%  0.00%   0 Policy Manager
  16           0         6          0  0.00%  0.00%  0.00%   0 DDR Timers
  17           0         2          0  0.00%  0.00%  0.00%   0 Entity MIB API
  18           8        24        333  0.00%  0.00%  0.00%   0 EEM ED Syslog
  19           0       442          0  0.00%  0.00%  0.00%   0 HC Counter Timer
```

Figura 24: Utilização de CPU no RotC com autenticação TCP MD5.
Fonte: autoria própria.

A utilização de CPU atingiu uma média total de 3%.

4 ANÁLISE DE RESULTADOS

A partir dos resultados obtidos nos cenários 1 e 2 e seus respectivos testes, nota-se que não houveram grandes diferenças de desempenho entre a configuração básica do protocolo BGP e a sua configuração com autenticação TCP MD5.

No primeiro teste, onde foi verificado o tempo de resposta entre os dispositivos host-A e host-F em uma situação ótima e também com uma sobrecarga de dados na rede, percebe-se que, na situação ótima, os tempos de resposta foram praticamente os mesmos, 2,736 ms no cenário 1 contra 2,763 ms no cenário 2. Entretanto, com a sobrecarga de dados na rede sendo gerada pelo *software* JPerf, houve um aumento no tempo médio de resposta de **21,66%**, o que pode ser melhor visualizado no gráfico da figura 25:

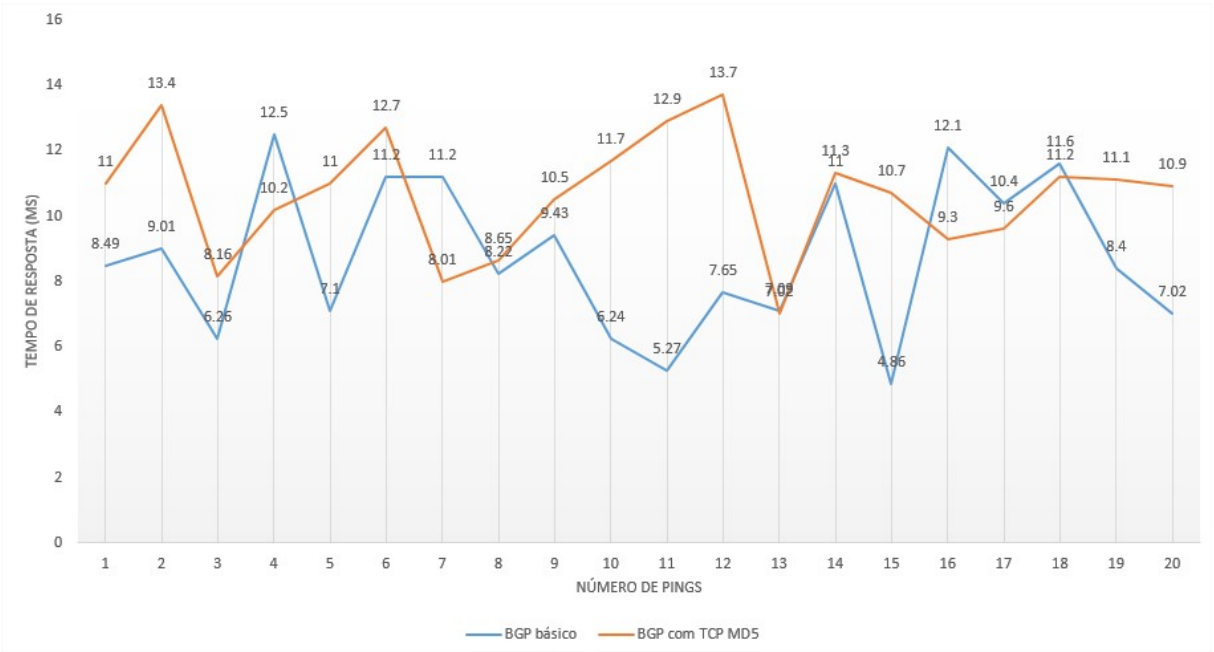


Figura 25: Gráfico comparativo do Teste 1.
Fonte: autoria própria.

Já no segundo teste, onde foi observado o tempo de convergência do protocolo

BGP após uma falha no enlace entre os roteadores RotC e RotD, percebe-se que não houve nenhuma diferença entre os cenários 1 e 2. Tanto na configuração básica do BGP quanto na sua configuração com autenticação TCP MD5 foram perdidos 30 dos 50 pacotes transmitidos até que a convergência para o enlace redundante fosse realizada. Esse resultado pode ser considerado plausível uma vez que a configuração de autenticação TCP MD5 não está relacionada ao cálculo de seleção das melhores rotas e à tabela de roteamento. Além disso, observa-se que em ambos os cenários as rotas utilizadas antes da queda do enlace eram as mesmas, o que mais uma vez indica a não relação entre a configuração de autenticação TCP MD5 e o algoritmo de seleção de rotas.

Por fim, no terceiro teste também foi observado que não houve nenhuma diferença entre os cenários 1 e 2 no que diz respeito a utilização de CPU do roteador RotC durante a geração de tráfego de dados entre os dispositivos host-B e host-E. Diferentemente do teste anterior, esse resultado não era esperado pois a configuração da autenticação TCP MD5 implica que o roteador, para cada segmento TCP relacionado ao protocolo BGP, terá que processar a função *hash* MD5 tendo como entrada a concatenação da mensagem BGP com a chave secreta, ao invés de simplesmente enviar a mensagem BGP sozinha. Entretanto, é provável que a limitação da largura de banda em 2 Mbit/s das interfaces seriais dos roteadores tenha feito com que o poder de processamento dos mesmos não fosse levado a um estresse máximo, fazendo com que não houvesse uma diferença perceptível no poder de processamento utilizado nos testes dos dois cenários.

5 CONSIDERAÇÕES FINAIS

O protocolo de roteamento *Border Gateway Protocol* tem sido o mais utilizado para a função de conectar os diversos sistemas autônomos presentes no mundo, o que é conhecido também como roteamento interdomínios. Ele vem desempenhando uma importante função no roteamento da Internet desde o início dos anos 90, ou seja, desde os primórdios da operação da Internet, proporcionando uma melhor infraestrutura de roteamento. O protocolo BGP evoluiu, substituindo o EGP e chegando na sua versão BGP-4.

Tendo isso em vista, nota-se a importância de se compreender os fundamentos deste protocolo, uma vez que ele é tão vital para o funcionamento da Internet. Ainda há muito a ser aprimorado quando se fala em roteamento na Internet e BGP, principalmente no que diz respeito à segurança do protocolo. Neste trabalho foram brevemente abordadas algumas das principais vulnerabilidades do protocolo BGP e uma simples forma de se mitigar algumas delas: a implementação da autenticação via TCP MD5 entre os diversos parceiros BGP. Embora seja sabido que a função *hash* MD5 não é a mais segura atualmente, já que existem outras funções como o MD6 e o SHA-2, ela ainda consegue mitigar certos tipos de ataques sem prejudicar em grandes proporções o desempenho do protocolo. Conforme visto neste trabalho, a implementação da autenticação via TCP MD5 aumentou em 21,66% o tempo de resposta na topologia de testes montada quando acometida por um alto tráfego de dados. Os demais testes realizados não mostraram diferenças significativas entre a configuração básica do BGP e a configuração com autenticação. Portanto, entende-se que o uso desta funcionalidade seja proveitoso.

Portanto, dada a importância do assunto, futuros trabalhos podem explorar outras frentes com o objetivo de se verificar o desempenho do protocolo BGP sob outras configurações de segurança. Por exemplo, pode-se aplicar *IP Security* (IPsec) ao tráfego BGP para protegê-lo a nível de camada de rede. Além disso, diferentes topologias podem ser aplicadas, especialmente topologias maiores onde haverá uma tabela de roteamento maior e mais próxima ao que temos na Internet hoje em dia.

REFERÊNCIAS

CAIS. **Vulnerabilidade no protocolo TCP que afetam BGP**. [S.l.], Abril 2004. Disponível em: <https://memoria.rnp.br/cais/alertas/2004/NISCC-236929.html>.

CISCO. **Introduction to EIGRP**. 2005. Disponível em: <http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13669-1.html>.

CISCO. **OSPF Design Guide**. 2005. Disponível em: <http://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html>.

CISCO. **Configuring Routing Information Protocol**. 2006. Disponível em: <http://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/configuration/guide/fipr_c/1cfrip.html>.

CISCO. **Cisco IOS BGP Configuration Guide, Release 12.4T - Configuring a Basic BGP Network**. 2007. Disponível em: <http://www.cisco.com/c/en/us/td/docs/ios/12_2sr/12_2srb/feature/guide/tbgp_c/t_brbbas.html>

CISCO. **Routing Information Protocol**. 2012. Disponível em: <http://docwiki.cisco.com/wiki/Routing_Information_Protocol>.

CISCO. **Enhanced Interior Gateway Routing Protocol**. 2015. Disponível em: <http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/16406-eigrp-toc.html>.

DOYLE, J. **Routing TCP/IP**. 1th. ed. [S.l.]: Cisco Press, 1998.

DUGAN, J. et al. **iPerf - The network bandwidth measurement tool**. Disponível em: <https://iperf.fr>.

HALABI, S.; MCPHERSON, D. **Internet Routing Architectures**. 2nd. ed. [S.l.]: Cisco Press, 2000.

HEFFERNAN, A. **Protection of BGP sessions via the TCP MD5 signature option**. [S.l.], Agosto 1998. 1-6 p. Disponível em: <https://tools.ietf.org/html/rfc2385>.

HUSTON, G. Exploring autonomous system numbers. **The Internet Protocol Journal**, v. 9, n. 1, 2006. Disponível em: <http://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-12/autonomous-system-numbers.html>.

IETF. **OSPF Version 2**. 1998. Disponível em: <https://www.ietf.org/rfc/rfc2328.txt>.

JAIN, V.; EDGEWORTH, B. **Troubleshooting BGP: A Practical Guide to Understanding and Troubleshooting BGP**. 1th. ed. [S.l.]: Cisco Press, 2016.

MOURA, A. S. D. Roteamento: O que é importante saber. **Revista NewsGeneration - RNP**, v. 1, n. 1, 1997. Disponível em: <https://memoria.rnp.br/newsgen/9705/n1-1.html>.

MOURA, A. S. D. O protocolo bgp4 - parte 1. **Revista NewsGeneration - RNP**, v. 3, n. 2, 1999. Disponível em: <http://memoria.rnp.br/newsgen/9903/bgp4.html>.

MOURA, A. S. D. Dicas na configuração do protocolo bgp-4 - parte 1. **Revista NewsGeneration - RNP**, v. 5, n. 1, 2001. Disponível em: <https://memoria.rnp.br/newsgen/0101/bgp4-dicas.html>.

MURPHY, S. **BGP Security Vulnerabilities Analysis**. [S.l.], January 2006. 1-22 p. Disponível em: <https://www.ietf.org/rfc/rfc4272.txt>.

PAQUET, C.; TEARE, D. **Construindo Redes Cisco Escaláveis**. 1st. ed. [S.l.]: Cisco Press, 2003.

REKHTER, Y.; LI, T.; HARES, S. **A Border Gateway Protocol 4 (BGP-4)**. [S.l.], January 2006. 1-105 p. Disponível em: <http://www.ietf.org/rfc/rfc4271.txt>.

STALLINGS, W. **Data and Computer Communications (8th Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2007. ISBN 0132433109.

TANENBAUM, A. S.; WETHERALL, D. J. **Computer networks**. 5th. ed. [S.l.]: PRENTICE HALL, 2011.

WANG, H.; WANG, C.; YU, G. Bgp security configuration in isp networks. **PIERS Proceedings**, 2009.

YIN, R. K. **Case study research: Design and methods**. [S.l.]: Sage publications, 2013.

APÊNDICE A – CONFIGURAÇÃO DO ROTEADOR ROTA

```
RotA#show running-config
Building configuration...

Current configuration : 1372 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RotA
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$36L6$98TuWdPxc1ITJgQ1QJ0Pd/
!
no aaa new-model
dot11 syslog
ip cef
!
no ip domain lookup
!
multilink bundle-name authenticated
!
archive
log config
hidekeys
```

```
!
interface FastEthernet0/0
ip address 10.1.1.1 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface Serial0/0/0
ip address 203.0.113.22 255.255.255.252
!
interface Serial0/1/0
ip address 203.0.113.1 255.255.255.252
no fair-queue
clock rate 2000000
!
router bgp 64512
no synchronization
bgp log-neighbor-changes
network 10.1.1.0 mask 255.255.255.0
network 10.10.1.0 mask 255.255.255.0
network 203.0.113.0 mask 255.255.255.252
network 203.0.113.20 mask 255.255.255.252
neighbor 203.0.113.2 remote-as 64516
neighbor 203.0.113.2 password tccbgppass
neighbor 203.0.113.21 remote-as 64514
neighbor 203.0.113.21 password tccbgppass
no auto-summary
!
ip forward-protocol nd
!
```

```
ip http server
no ip http secure-server
!
control-plane
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
end
```

APÊNDICE B – CONFIGURAÇÃO DO ROTEADOR ROTB

```
RotB#show running-config
Building configuration...

Current configuration : 1139 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RotB
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$9SJP$xyx4v./9jIi/8gwGDJMT3.
!
no aaa new-model
ip cef
!
no ip domain lookup
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
interface FastEthernet0/0
ip address 10.2.2.1 255.255.255.0
duplex auto
speed auto
```

```
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
!
interface Serial0/1/0
ip address 203.0.113.5 255.255.255.252
no fair-queue
clock rate 2000000
!
router bgp 64513
no synchronization
bgp log-neighbor-changes
network 10.2.2.0 mask 255.255.255.0
network 10.10.2.0 mask 255.255.255.0
network 203.0.113.4 mask 255.255.255.252
neighbor 203.0.113.6 remote-as 64516
neighbor 203.0.113.6 password tccbgppass
no auto-summary
!
ip forward-protocol nd
!
ip http server
no ip http secure-server
!
control-plane
!
line con 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
end
```

APÊNDICE C – CONFIGURAÇÃO DO ROTEADOR ROTC

```
RotC#show running-config
Building configuration...

Current configuration : 1353 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RotC
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$PdlE$SLK50pxW7/prMyqlQpKnB1
!
no aaa new-model
!
resource policy
!
ip cef
!
no ip domain lookup
!
voice-card 0
!
interface FastEthernet0/0
```

```
no ip address
shutdown
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface Serial0/1/0
ip address 203.0.113.6 255.255.255.252
no fair-queue
!
interface Serial0/1/1
ip address 203.0.113.2 255.255.255.252
!
interface Serial0/3/0
ip address 203.0.113.9 255.255.255.252
clock rate 2000000
!
router bgp 64516
no synchronization
bgp log-neighbor-changes
network 203.0.113.0 mask 255.255.255.252
network 203.0.113.4 mask 255.255.255.252
network 203.0.113.8 mask 255.255.255.252
neighbor 203.0.113.1 remote-as 64512
neighbor 203.0.113.1 password tccbgppass
neighbor 203.0.113.5 remote-as 64513
neighbor 203.0.113.5 password tccbgppass
neighbor 203.0.113.10 remote-as 64516
neighbor 203.0.113.10 password tccbgppass
no auto-summary
```

```
!
ip http server
no ip http secure-server
!
control-plane
!
line con 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
end
```

APÊNDICE D – CONFIGURAÇÃO DO ROTEADOR ROTD

```
RotD#show running-config
Building configuration...

Current configuration : 1546 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RotD
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$hDTH$4rQJuQ6.4dzVFHI/iEsgb.
!
no aaa new-model
dot11 syslog
ip cef
!
no ip domain lookup
multilink bundle-name authenticated
!
voice-card 0
!
archive
log config
```

```
hidekeys
!
interface FastEthernet0/0
no ip address
shutdown
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface Serial0/1/0
ip address 203.0.113.17 255.255.255.252
no fair-queue
clock rate 2000000
!
interface Serial0/1/1
ip address 203.0.113.13 255.255.255.252
clock rate 2000000
!
interface Serial0/3/0
ip address 203.0.113.10 255.255.255.252
!
router bgp 64516
no synchronization
bgp log-neighbor-changes
network 203.0.113.8 mask 255.255.255.252
network 203.0.113.12 mask 255.255.255.252
network 203.0.113.16 mask 255.255.255.252
neighbor 203.0.113.9 remote-as 64516
neighbor 203.0.113.9 password tccbgppass
neighbor 203.0.113.14 remote-as 64514
```

```
neighbor 203.0.113.14 password tccbgppass
neighbor 203.0.113.18 remote-as 64515
neighbor 203.0.113.18 password tccbgppass
no auto-summary
!
ip forward-protocol nd
!
ip http server
no ip http secure-server
!
control-plane
!
voice-port 0/0/0
!
voice-port 0/0/1
!
voice-port 0/0/2
!
voice-port 0/0/3
!
line con 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
end
```

APÊNDICE E – CONFIGURAÇÃO DO ROTEADOR ROTE

```
RotE#show running-config
Building configuration...

Current configuration : 1263 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RotE
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$ZiAt$I2edkki5jkpxa7tebcG0O.
!
no aaa new-model
!
resource policy
!
ip cef
!
no ip domain lookup
!
interface FastEthernet0/0
ip address 10.3.3.1 255.255.255.0
duplex auto
```

```
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface Serial0/0/0
ip address 203.0.113.21 255.255.255.252
clock rate 2000000
!
interface Serial0/1/0
ip address 203.0.113.14 255.255.255.252
no fair-queue
!
router bgp 64514
no synchronization
bgp log-neighbor-changes
network 10.3.3.0 mask 255.255.255.0
network 10.10.3.0 mask 255.255.255.0
network 203.0.113.12 mask 255.255.255.252
network 203.0.113.20 mask 255.255.255.252
neighbor 203.0.113.13 remote-as 64516
neighbor 203.0.113.13 password tccbgppass
neighbor 203.0.113.22 remote-as 64512
neighbor 203.0.113.22 password tccbgppass
no auto-summary
!
ip http server
no ip http secure-server
!
control-plane
!
line con 0
```

```
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
end
```

APÊNDICE F – CONFIGURAÇÃO DO ROTEADOR ROTF

```
RotF#show running-config
Building configuration...

Current configuration : 1191 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RotF
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$iy7B$k8X78wIfdmLFlbDJ80qWo0
!
no aaa new-model
!
resource policy
!
ip cef
!
no ip domain lookup
!
interface FastEthernet0/0
ip address 10.4.4.1 255.255.255.0
duplex auto
```

```
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface Serial0/0/0
no ip address
shutdown
no fair-queue
clock rate 2000000
!
interface Serial0/1/0
no ip address
shutdown
clock rate 2000000
!
interface Serial0/1/1
ip address 203.0.113.18 255.255.255.252
!
router bgp 64515
no synchronization
bgp log-neighbor-changes
network 10.4.4.0 mask 255.255.255.0
network 10.10.4.0 mask 255.255.255.0
network 203.0.113.16 mask 255.255.255.252
neighbor 203.0.113.17 remote-as 64516
neighbor 203.0.113.17 password tccbgppass
no auto-summary
!
ip http server
no ip http secure-server
!
```

```
control-plane
!
line con 0
line aux 0
line vty 0 4
login
!
scheduler allocate 20000 1000
end
```