

Redes de COmputadores II

UNIVERSIDADE DO MINHO | Mestrado integrado em telecomunicações e informática

**Grupo 1**

Relatório Final do Trabalho Prático

Constituição do grupo



**Nome:** André Machado

**Número Mecanográfico:** A66693

**E-mail:** andre.machado93@gmail.com

**Correio eletrónico institucional:** a66693@alunos.uminho.pt



**Nome:** Carlos Carvalho

**Número Mecanográfico:** A75401

**E-mail:** carlosc.96@hotmail.com

**Correio eletrónico institucional:** a75401@alunos.uminho.pt



**Nome:** Rui Silva

**Número Mecanográfico:** A69987

**E-mail:** rui.silva.kj@gmail.com

**Correio eletrónico institucional:** a69987@alunos.uminho.pt

Índice Geral

[Constituição do grupo 2](#_Toc484968084)

[Índice Geral 3](#_Toc484968085)

[Índice de Figura 3](#_Toc484968086)

[Índice de Tabela 4](#_Toc484968087)

[1.1. Introdução 5](#_Toc484968088)

[1.2. Análise do problema 6](#_Toc484968089)

[1.3. Implementação do projeto 8](#_Toc484968090)

[1.3.1. Protocolo OSPF 8](#_Toc484968091)

[1.3.1.1. Configuração do protocolo OSPF 9](#_Toc484968092)

[1.3.2. Protocolo RIP 14](#_Toc484968093)

[1.3.2.1. Configuração do RIP 14](#_Toc484968094)

[1.3.3. Configuração do endereçamento estático 17](#_Toc484968095)

[1.3.4. Configuração das Politicas BGP 18](#_Toc484968096)

[1.3.4.1. AS65000 18](#_Toc484968097)

[1.3.4.2. AS65100 19](#_Toc484968098)

[1.3.4.3. AS65200 26](#_Toc484968099)

[1.3.4.4. AS64300 27](#_Toc484968100)

[1.3.4.5. AS65400 30](#_Toc484968101)

[1.3.4.6. Conclusão da Análise das Tabelas 31](#_Toc484968102)

Índice de Figura

[Figura 1 – Topologia Geral. 7](#_Toc484943908)

[Figura 2 – Implementação do OSPF na nossa topologia. 9](#_Toc484943909)

[Figura 3 - Running-config router n5. 10](#_Toc484943910)

[Figura 4 - Show ip ospf do router n5. 11](#_Toc484943911)

[Figura 5 – Show running-config e show ip OSPF do router n10. 11](#_Toc484943912)

[Figura 6 – Show running-config do router n29. 12](#_Toc484943913)

[Figura 7 – Show ip OSPF do router 29. 12](#_Toc484943914)

[Figura 8 - Show running-config do router n26. 13](#_Toc484943915)

[Figura 9 - Show ip OSPF do router n26. 13](#_Toc484943916)

[Figura 10 - Implementação do RIP na nossa topologia. 14](#_Toc484943917)

[Figura 11 - Show Running Config R1 18](#_Toc484943918)

[Figura 12 - Show IP BGP R1 19](#_Toc484943919)

[Figura 13 - Show Running Config R2 20](#_Toc484943920)

[Figura 14 - Show Running Config R3 22](#_Toc484943921)

[Figura 15 - SHow Running Config R4 23](#_Toc484943922)

[Figura 16 - Show IP BGP R2 24](#_Toc484943923)

[Figura 17- Show IP BGP R3 24](#_Toc484943924)

[Figura 18 - Show IP BGP R4 25](#_Toc484943925)

[Figura 19 - Show Running Config R7 26](#_Toc484943926)

[Figura 20 - Show IP BGP R7 26](#_Toc484943927)

[Figura 21 - Show Running Config R5 28](#_Toc484943928)

[Figura 22Show Running Config R6 29](#_Toc484943929)

[Figura 23 - Show IP BGP R5 29](#_Toc484943930)

[Figura 24 - Show IP BGP R6 30](#_Toc484943931)

[Figura 25 - Show Running Config R8 31](#_Toc484943932)

[Figura 26 - Show IP BGP R8 31](#_Toc484943933)

# Introdução

No âmbito da Unidade Curricular de Redes de Computadores II do curso Mestrado Integrado em Engenharia de Telecomunicações e Informática foi proposto aos alunos o desenvolvimento e simulação de uma topologia na ferramenta de emulação Common Open Research Emulator (CORE).

Esta topologia visa por em prática todos conhecimentos que se adquiriu na unidade curricular ao longo do semestre, de forma a que nos possibilite uma melhor aprendizagem sobre todas etapas e problemas quando se está a configurar os diferentes protocolos e políticas. Este projeto está principalmente focado sobre a implementação de soluções de encaminhamento intra e inter-dominio, configuração do encaminhamento interno estático, RIP e OSPF, assim como encaminhamento externo BGP.

Este projeto também difere pela positiva pois é nos exigido trabalhar e aplicar diferentes políticas de encaminhamento.

Desta forma, inicialmente o grupo pretende falar sobre as diferentes configurações do OSPF, RIP e do routing estático. Posteriormente será elaborada uma identificação dos Area Border Routers (ABR) do OSPF na nossa topologia desenvolvida e fazer uma explicação das escolhas efetuadas. De seguida será elaborada uma explicação das diferentes politicas de encaminhamento externo do BGP que se utilizou na nossa topologia. Recorreu-se ao uso de exemplos das tabelas de encaminhamento e comentários para uma melhor explicação e ainda por fim falar das diferentes escolhas da topologia entre sistemas autónomos.

# Análise do problema

A topologia proposta está dividida em 5 sistemas autónomos diferentes que têm que ter conetividade global entre eles, estando sobre o efeito de diversas politicas de encaminhamento. O sistema autónomo (AS) **65000**, é um sistema autónomo stub que mantém relações de peering BGP que lhe garante acesso externo, porém internamente utiliza o o protocolo de encaminhamento RIP. Este terá a gama de IPv4 10.0.0.0/16. Este AS é composto por um único border router através do qual faz ligação com o seu ISP, o sistema autónomo 65200 e com o seu parceiro 65100, que tem um contrato de parceria que lhe permite encaminhar tráfego diretamente para ele.

O sistema autónomo **65100** é um sistema multihomed, que utiliza internamente a gama de IP 10.1.0.0/16 e o protocolo de encaminhamento OSPF, porém mantém relações de peering com o BGP exterior com o vizinho AS 65200 e 65300. Internamente este AS tem 3 borders routers que mantém entre eles ligações i-BGP sendo 2 deles e-BGP. Este sistema autónomo tem ainda a particularidade de encaminhar trafego entre o AS 65200 (ligação backup) e o 65100 (ligação preferencial) e não poderá aceitar o tráfego do AS65300. Para tingir o sistema autónomo 65400 terá que seguir pelo router R4 para AS65200 e posteriormente utilizar o AS6300, para entrar terá que ser pelo router R3.

O sistema autónomo **65200** é um sistema autónomo essencialmente de trânsito que terá de ter internamente os endereços da gama 10.2.0.0/16 e internamente o seu encaminhamento será estático. Este como já foi referido é o ISP dos sistemas autónomos 65100 e 65200 tendo diversas restrições a nível de fluxo como já evidenciado.

O sistema autónomo **65300** é um AS de transito. Este é também ISP do AS 65100. Este AS não deverá anunciar as rotas do AS 65000 ao AS 65400 pois este não pode usar o AS65300 para ter acesso ao AS65000.

O AS **65400** é um sistema autónomo de transito que tem uma gama de IPs compreendida em 10.4.0.0/16. Este tem apenas duas rotas, sendo a rota que este prefere para chegar ao AS 65100, é via 65300.

Deste modo a topologia tomou a seguinte forma:

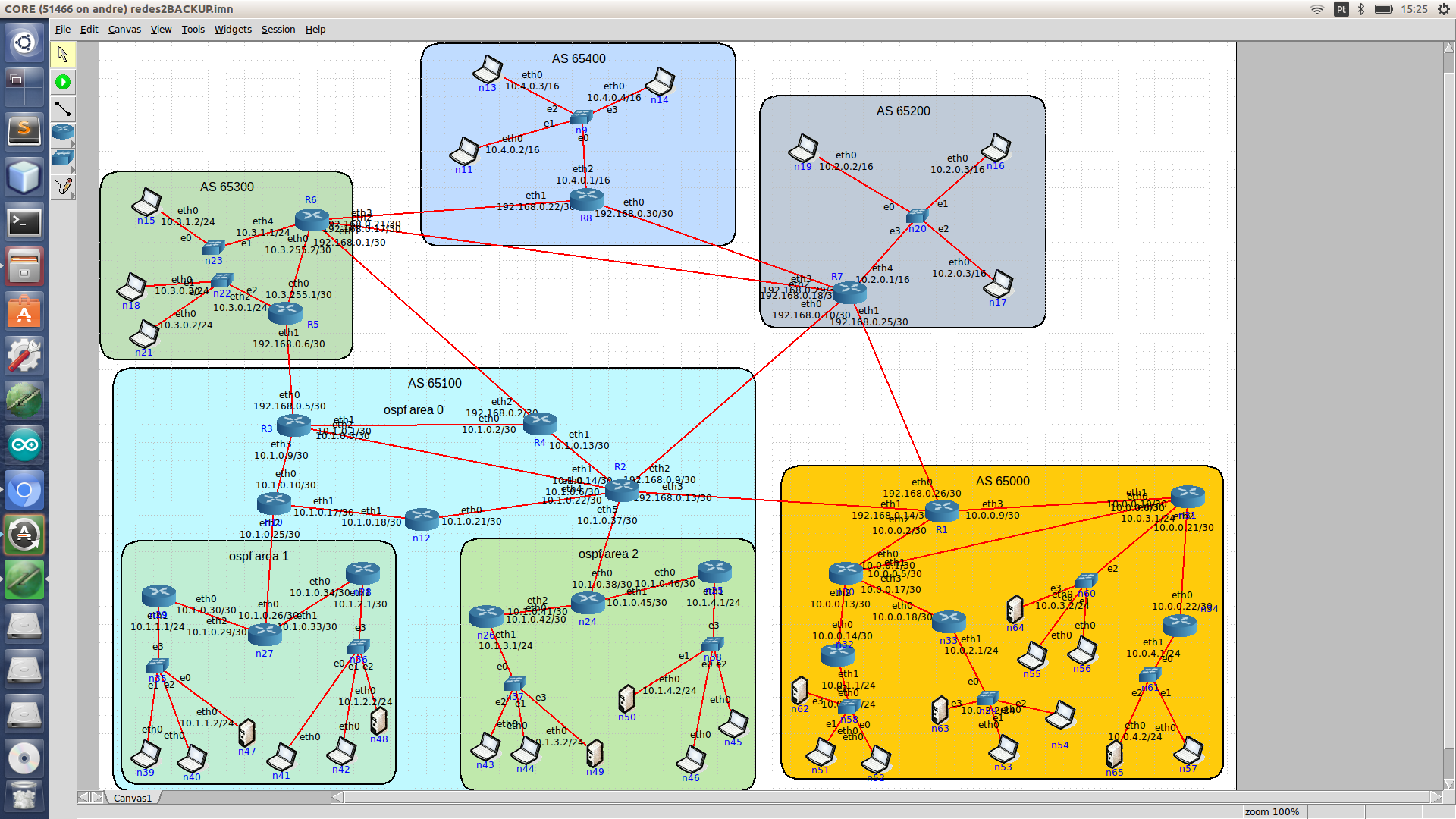


Figura – Topologia Geral.

# Implementação do projeto

## Protocolo OSPF

O protocolo OSPF é um protocolo de encaminhamento que foi implementado de forma a fazer face a falhas do protocolo RIP. Este protocolo tem como principais características a necessidade do conhecimento completo da topologia para calcular o caminho mais curto para chegar ao recetor. Com este protocolo é substituído uma tabela de encaminhamento estático por uma tabela que contém informação sobre todas ligações da rede e que é atualizada constantemente.

Este protocolo visa ter maior atenção sobre o valor de custo de uma ligação recorrendo ao uso de algoritmos de estado de ligação (LSAs). Desta forma este está diretamente relacionado com a largura de banda. Quanto maior a largura de banda menor vai ser o custo da ligação.

Este protocolo apresenta diversas caraterísticas, tais como a existência de 3 base de dados OSPF que são:

* **Base de dados de adjacências** – tabela com todos os vizinhos com quem o router estabeleceu ligação;
* **Base de dados de Link State Database** – base de dados topológica com o mapa da rede;
* **Base de dados de reenvio** – onde estão contidas as melhores rotas calculadas para cada destino.

Ter estas bases dados permite que haja uma convergência mais eficiente e mais rápida do algoritmo, porém pode se tornar uma desvantagem pois como esta tem que ter armazenada informação e conhecimento da topologia inteira pode necessitar de um uso excessivo de recursos computacionais (CPU e memória).

Este protocolo comunica sobretudo sobre 5 tipos de mensagens:

* **Type 1: helho** – é enviado com uma frequência elevada de forma a manter uma relação ativa com os vizinhos;
* **Type 2: Database Description** – sumário do conteúdo da base de dados com as ligações;
* **Type 3: Link-State Request** - pedido mais especifico sobre o link da base de dados de LSA do vizinho;
* **Type 4: Link-State Update** - Transporta LSA para os vizinhos;
* **Type 5: Link-State Acknowledgement** – confirma a receção de um LSA.

### Configuração do protocolo OSPF

Assim sendo implementou-se o protocolo OSPF na nossa topologia ficando da seguinte forma:

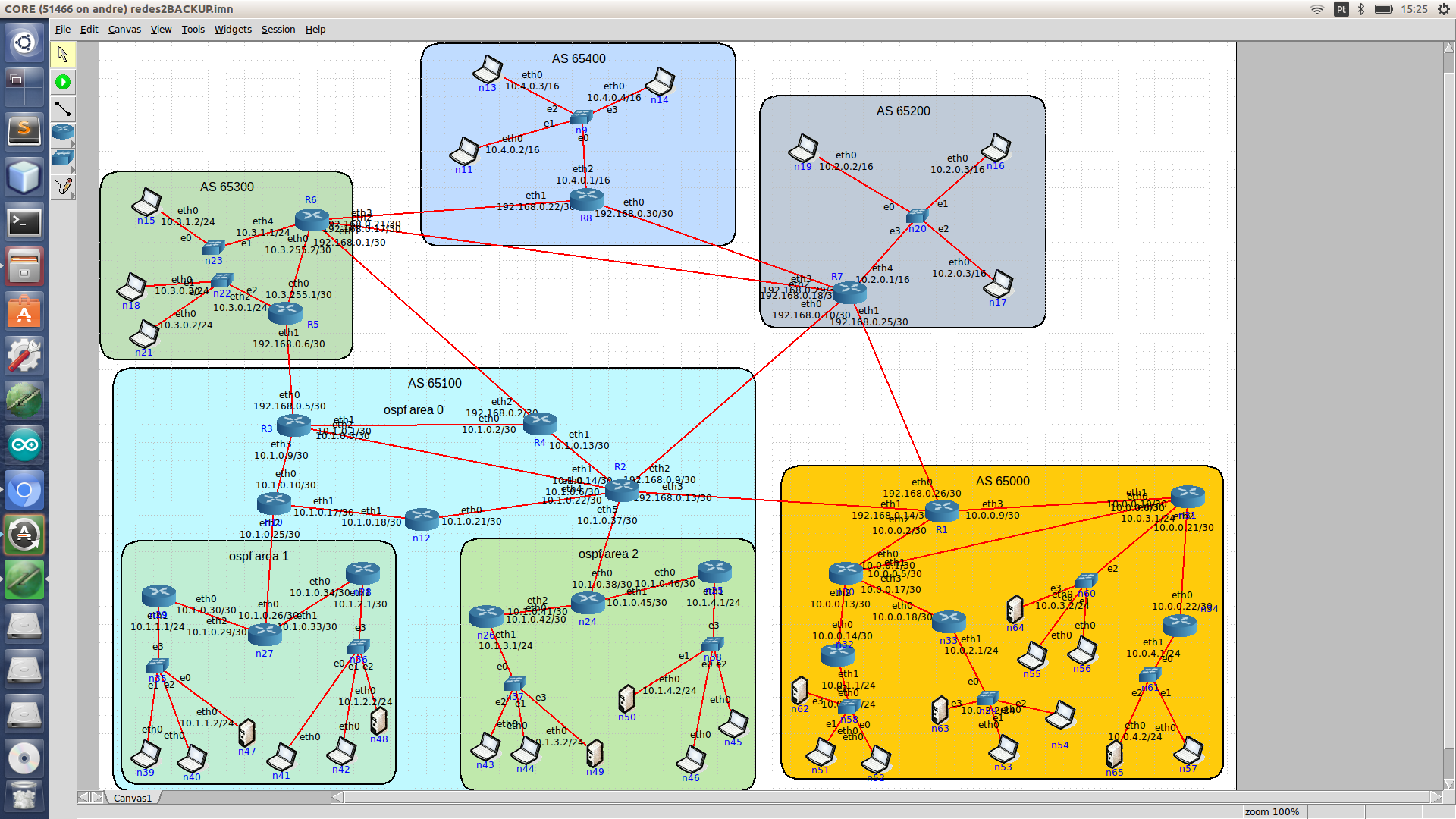


Figura – Implementação do OSPF na nossa topologia.

Dividiu-se a rede em 3 áreas:

* Área 0;
* Área 1;
* Área 2;

Uma área é uma coleção lógica de redes do OSPF, routers e ligações com a mesma identificação da área. Cada router tem que manter uma base de dados topológica para cada área a que pertence. Os routers assim não têm a necessidade de ter especificada as áreas do exterior, o possibilita uma diminuição da base de dados topológica de cada router.

Também desta forma, as áreas vão fazer com que aumente o nível de controlo e permite ainda uma melhor estabilidade.

Na área 0 está a nossa área backbone, que funciona como uma espécie de hub onde faz a distribuição da informação routing entre as diferentes áreas. Desta forma as diferentes áreas têm que estar ligadas sempre a esta área.

Para configurar o OSPF utilizou-se os seguintes comandos:

* Command: **router ospf:** ativa o OSPF num determinado router;
* OSPF Command: **ospf router-id a.b.c.d**: define um id para o router. O ID pode ser um endereço IP desse router, porém não é obrigatório. Obrigatório sim é não existir dois routers no mesmo domínio com ids iguais;
* OSPF Command: **network a.b.c.d/m area a.b.c.d:** Este comando especifica que interfaces do router tem o OSPF ativo e também a que área a rede dessa interface pertence.
* OSPF Command; **Redistribute BGP:** indica ao router para anunciar as redes atribuídas por BGP;
* Interface Command: **ip ospf authentication message-digest**: especifica que na interface que estamos a configurar é utilizada uma autenticação MD5 HMAC.
* Interface Command: **ip ospf message-digest-key KEYID md5 KEY:** define qual é a password utilizada (KEY) e encripta-a usando o MD5;

Configuração do BackBone:

* Router n5

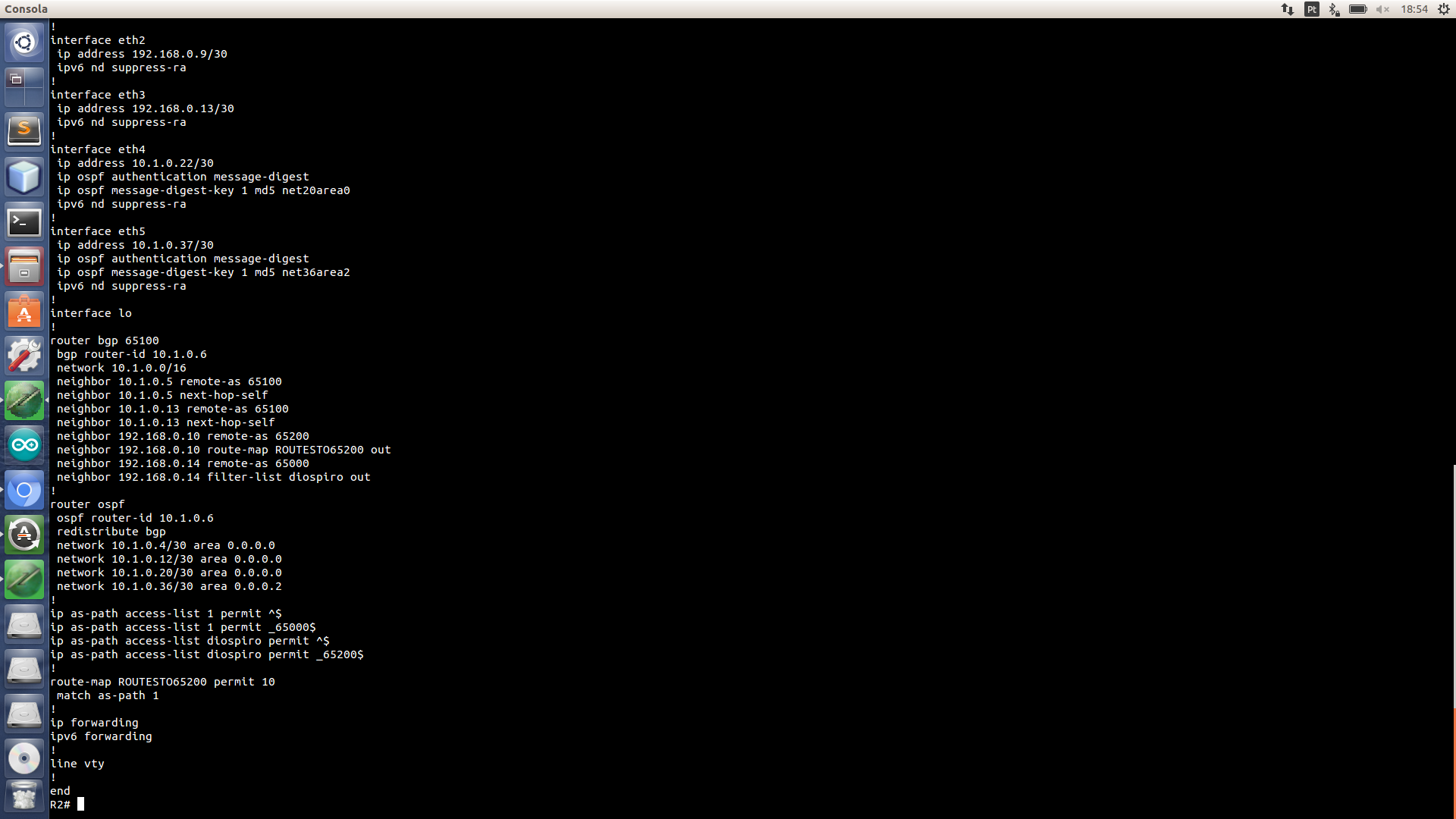
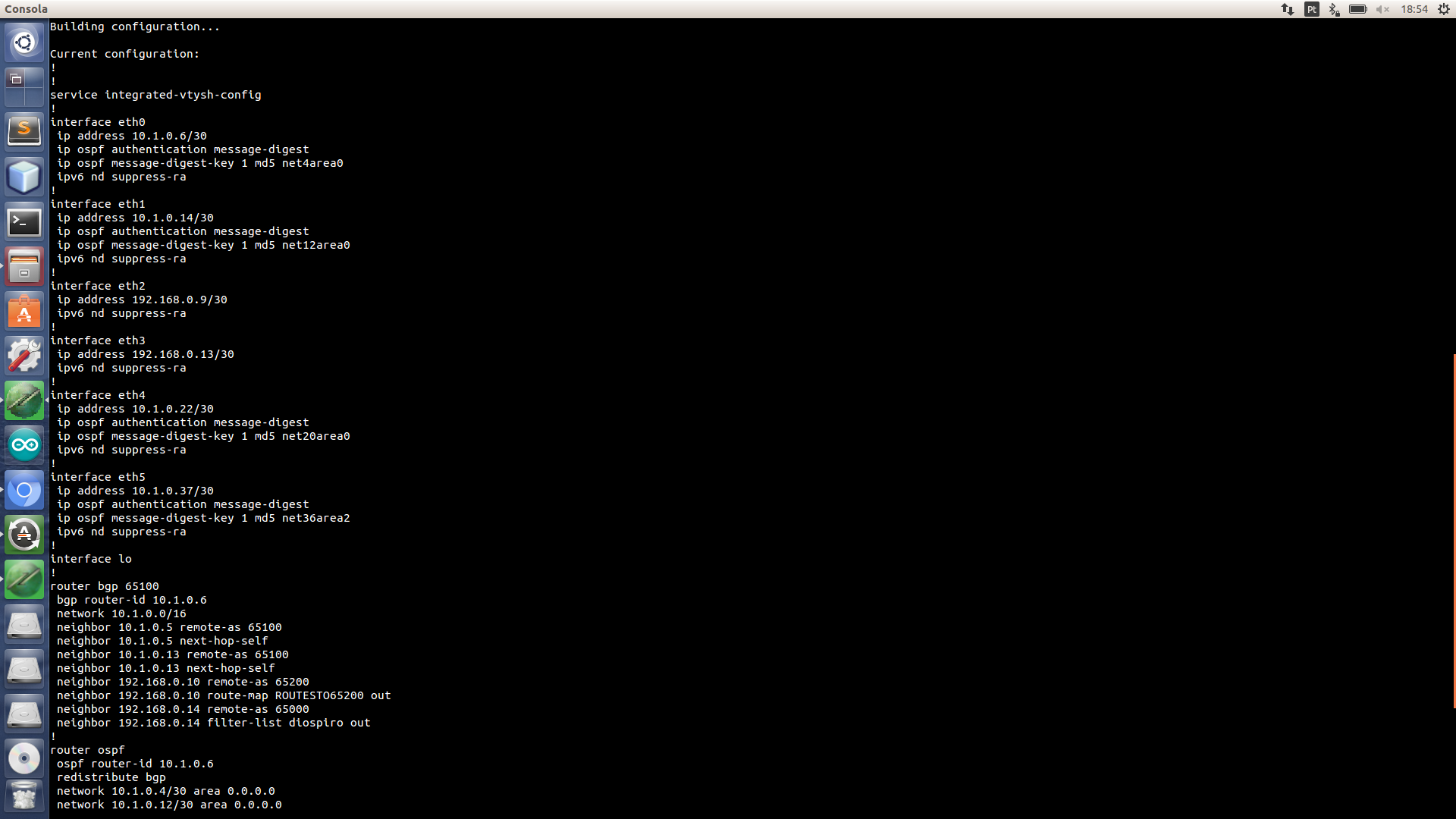


Figura - Running-config router n5.

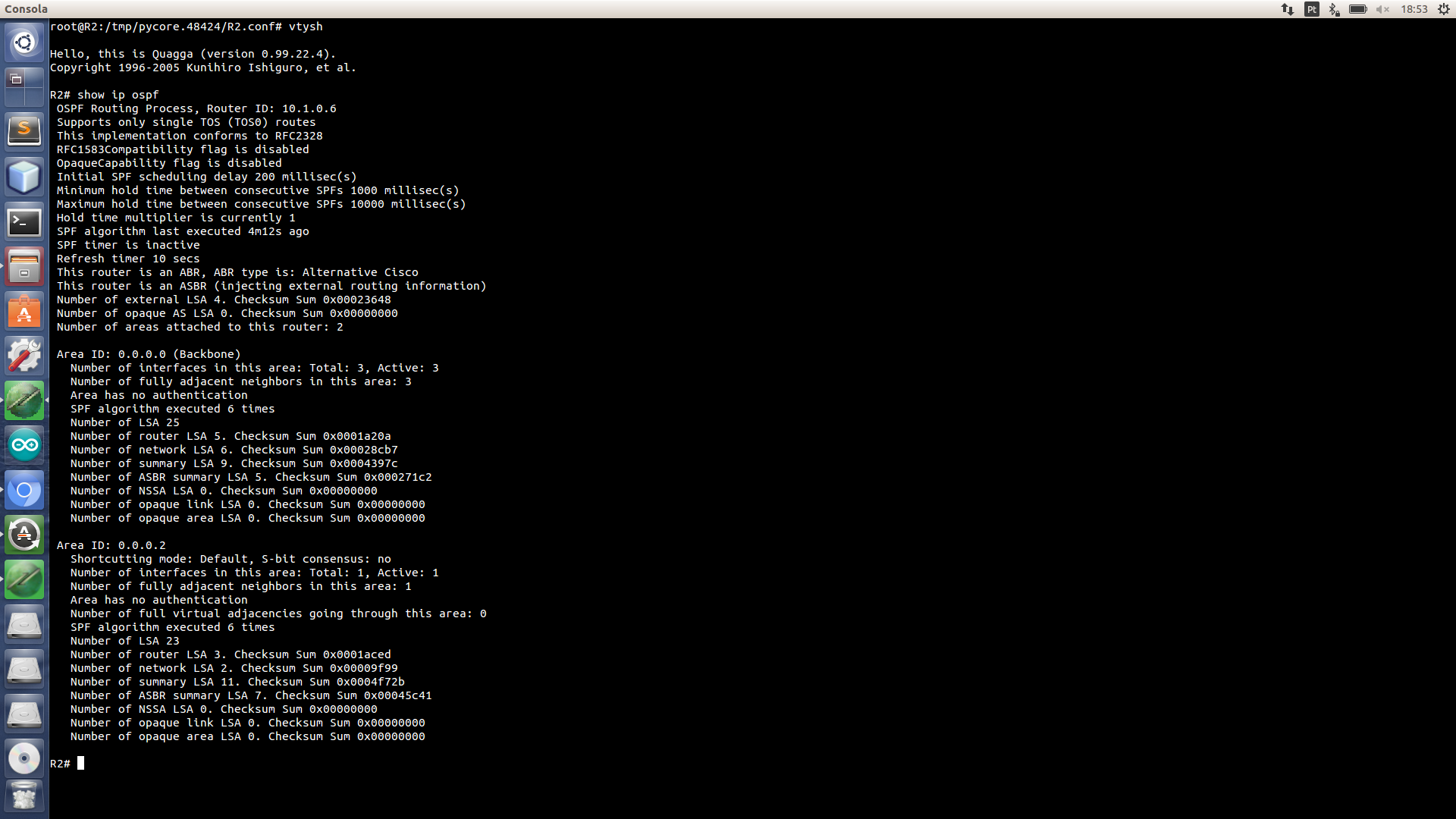


Figura - Show ip ospf do router n5.

* Router n10

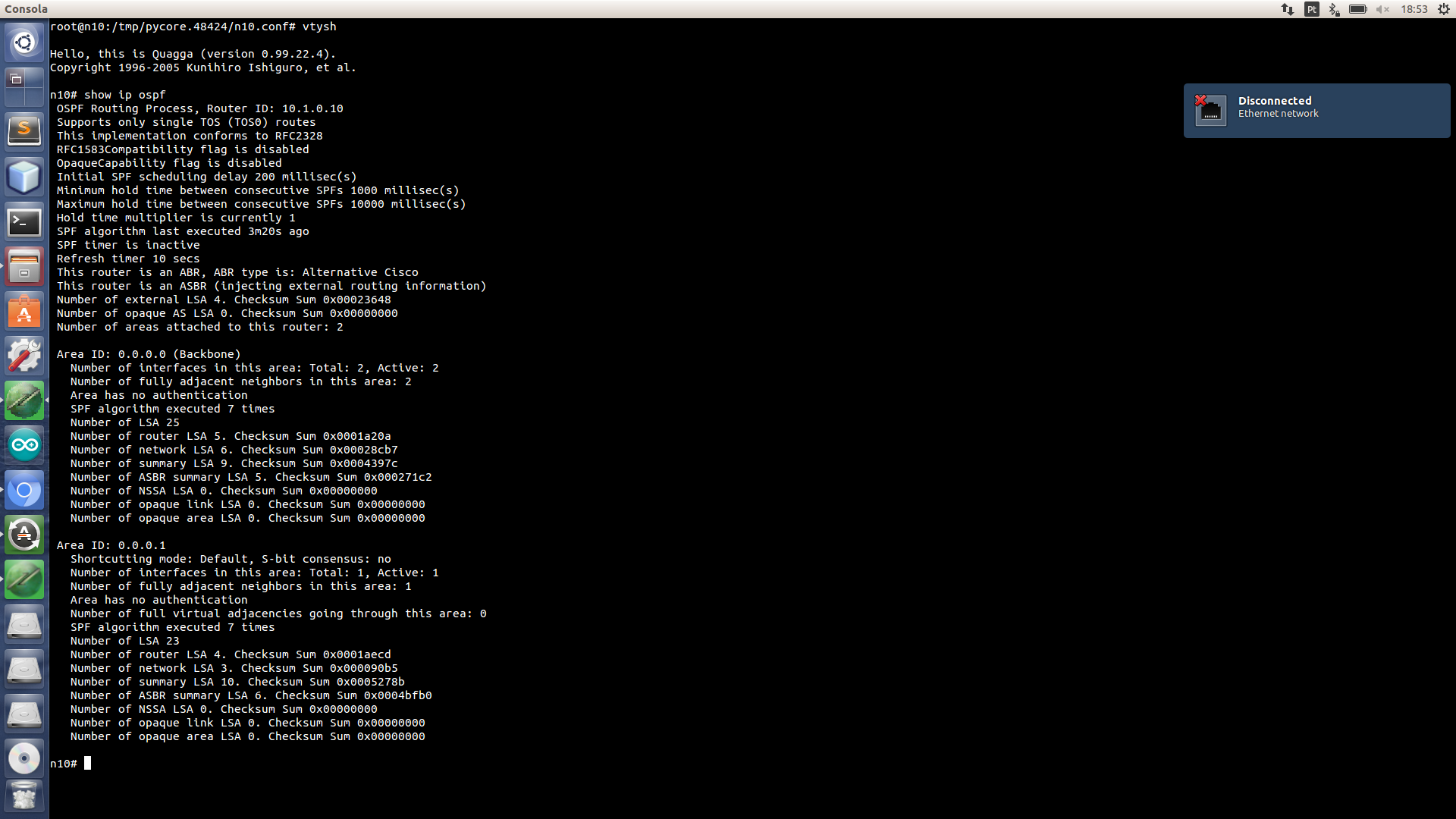
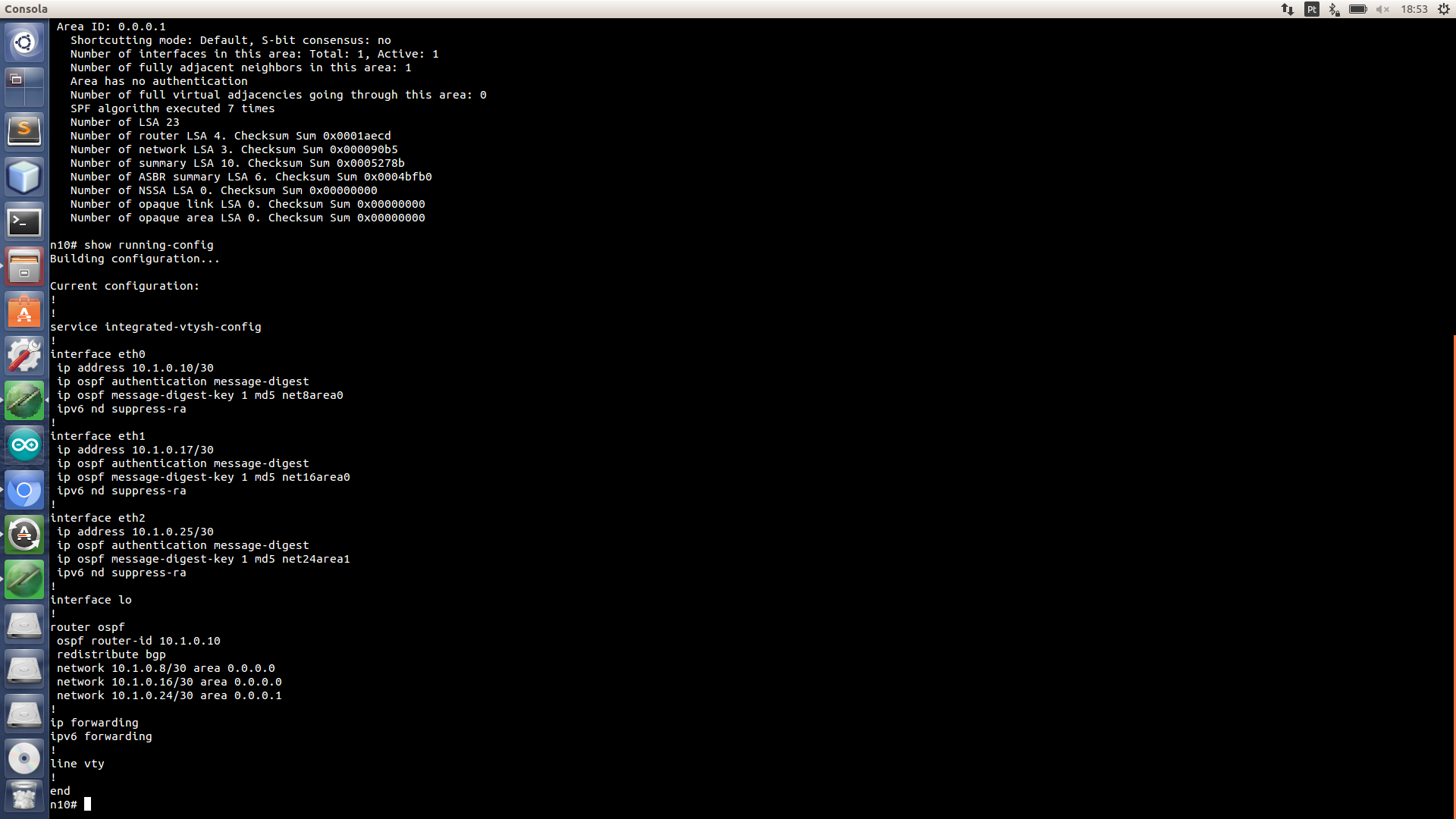


Figura – Show running-config e show ip OSPF do router n10.

* Router n29

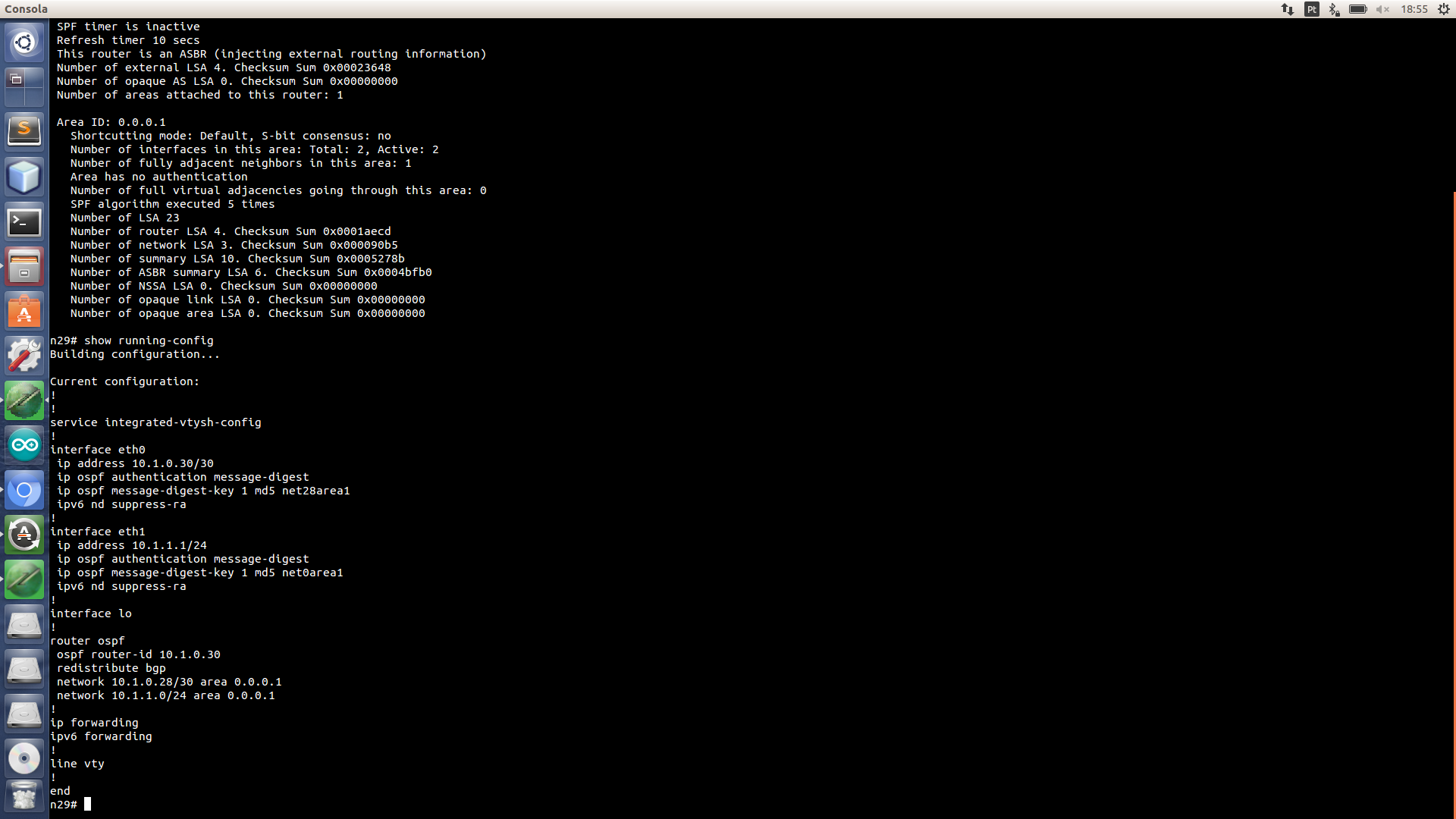


Figura – Show running-config do router n29.



Figura – Show ip OSPF do router 29.

* Router n26

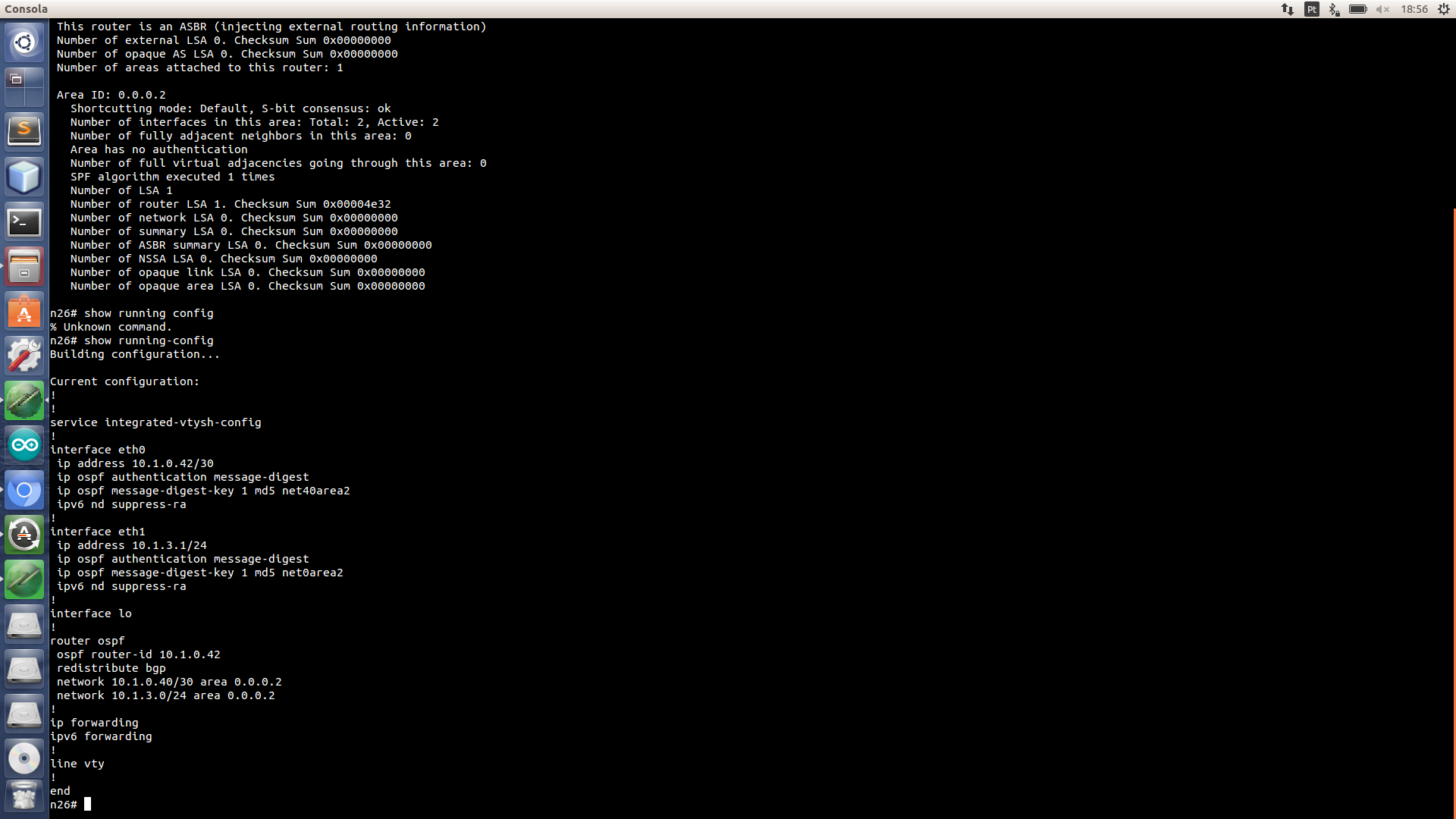


Figura - Show running-config do router n26.

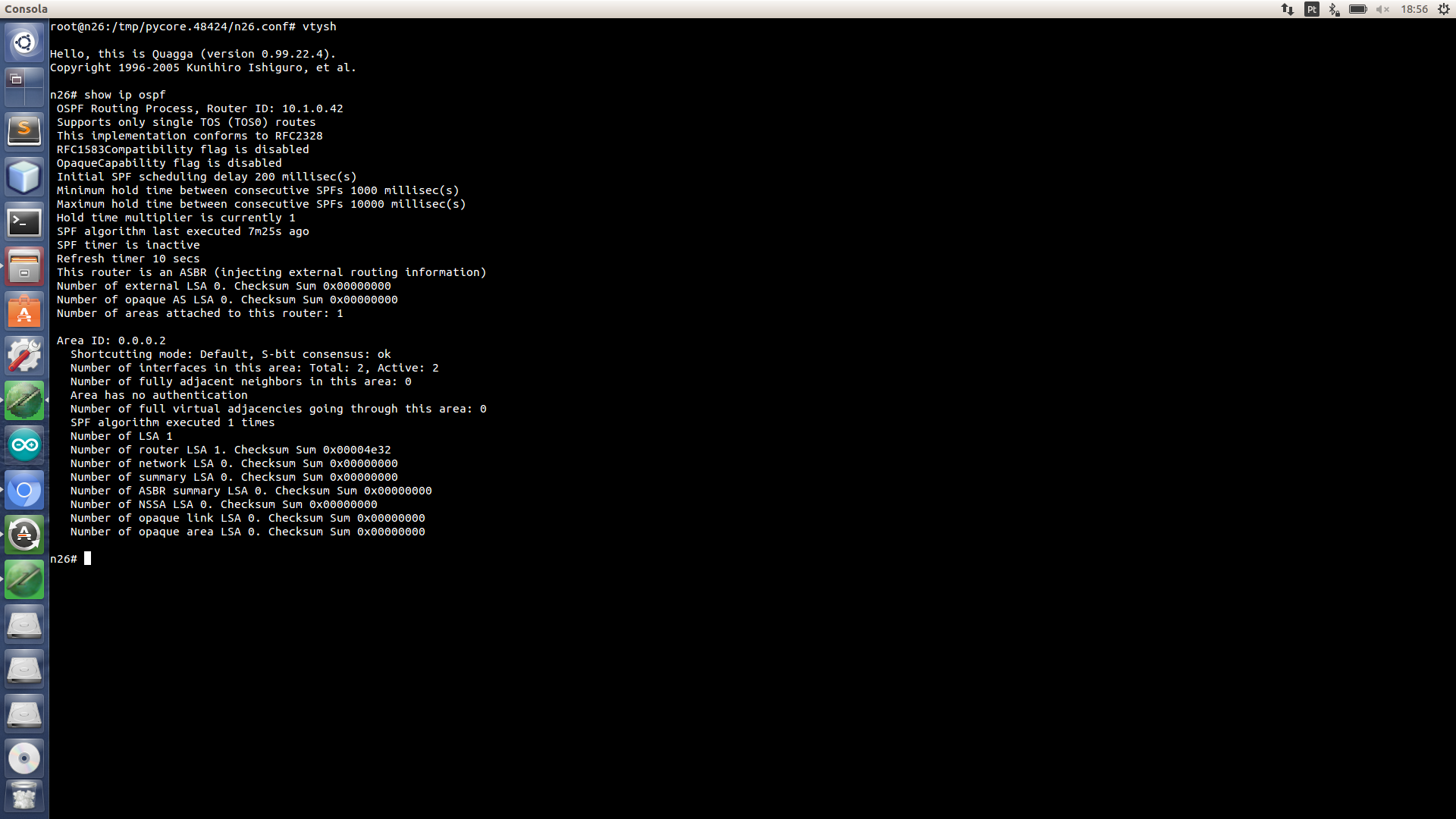


Figura - Show ip OSPF do router n26.

## Protocolo RIP

O protocolo de encaminhamento Routing Information Protocol (RIP) é um protocolo fácil implementação, que utiliza o algoritmo de vetor distância, usando como métrica para seleção da melhor rota o número de saltos entre a origem e destino.

Cada entidade envia periodicamente a todos os seus vizinhos um conjunto de mensagens com a atualização das de rotas e das novas distâncias.

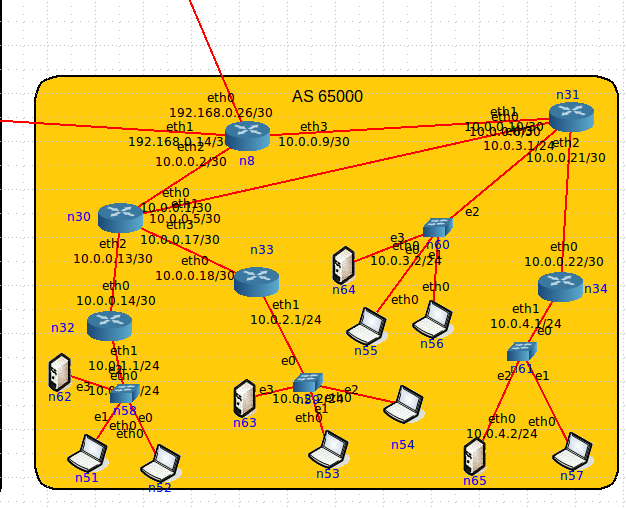


Figura - Implementação do RIP na nossa topologia.

### Configuração do RIP

Para configurar a área de acordo com os requisitos do projeto, o grupo fez uma pesquisa sobre os diversos comandos existentes para configurar o RIP nos routers pretendidos e fez o seguinte levantamento que posteriormente utilizou:

* Command: **router rip:** serve para ativar o RIP num determinado router. Este comando é necessário para fazer configurações posteriormente;
* Command: **no router rip:** serve para desativar o RIP num determinado router caso assim o pretendermos;
* RIP Command: **network network:** Serve para ativar o RIP na interface que pertente à network. A interface que coincide com a network fica assim com o RIP ativado;
* RIP Command: **neighbor a.b.c.d:**serve para especificar um RIP vizinho;
* RIP command: **redistribute bgp:** serve para indicar ao router para passar a informação referente a rotas aprendidas por BGP;
* Interface command: **ip rip authentication key-chain** *key-chain: coloca uma interface com uma autenticação key-chain;*

Após a configuração dos diversos routers RIP, as configurações resultantes foi a seguinte:

* No router R1



Figura - Running-config do router R1.

* No router n32

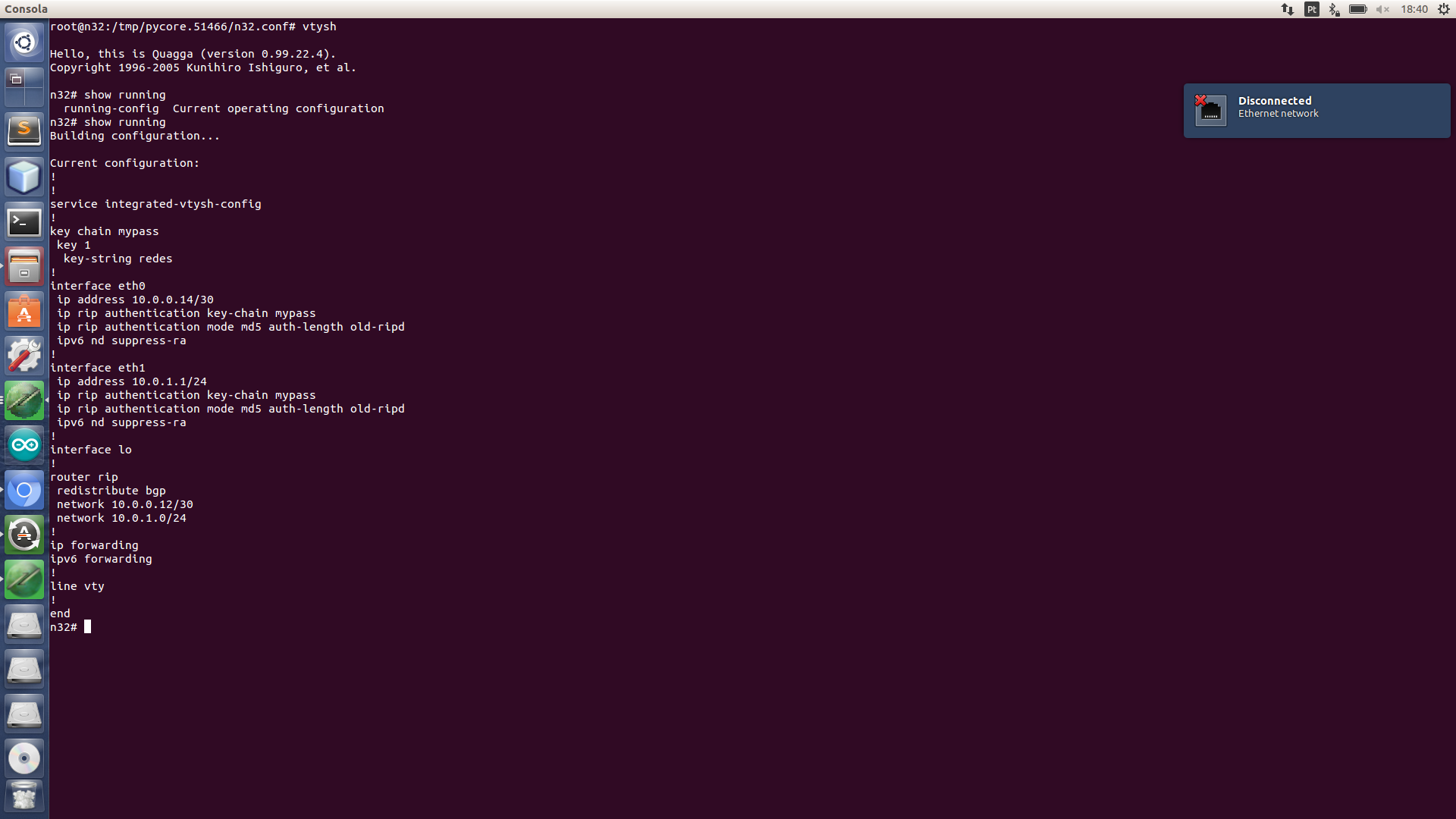


Figura - Show running-config router n32.

### Testes

De seguida, fez-se correr a topologia e os routers aprenderam as seguintes rotas. (show ip rip)

* O router R1

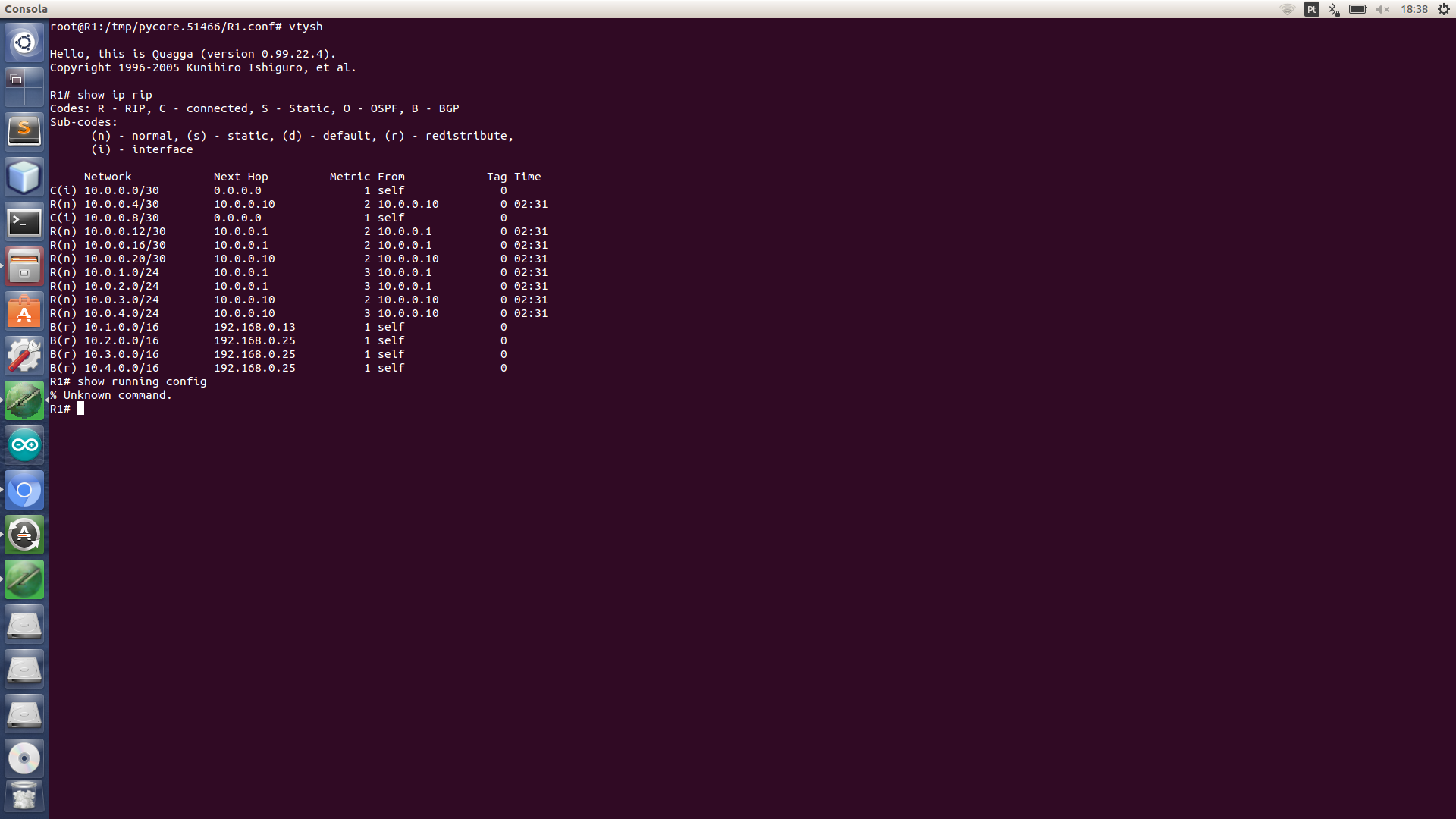
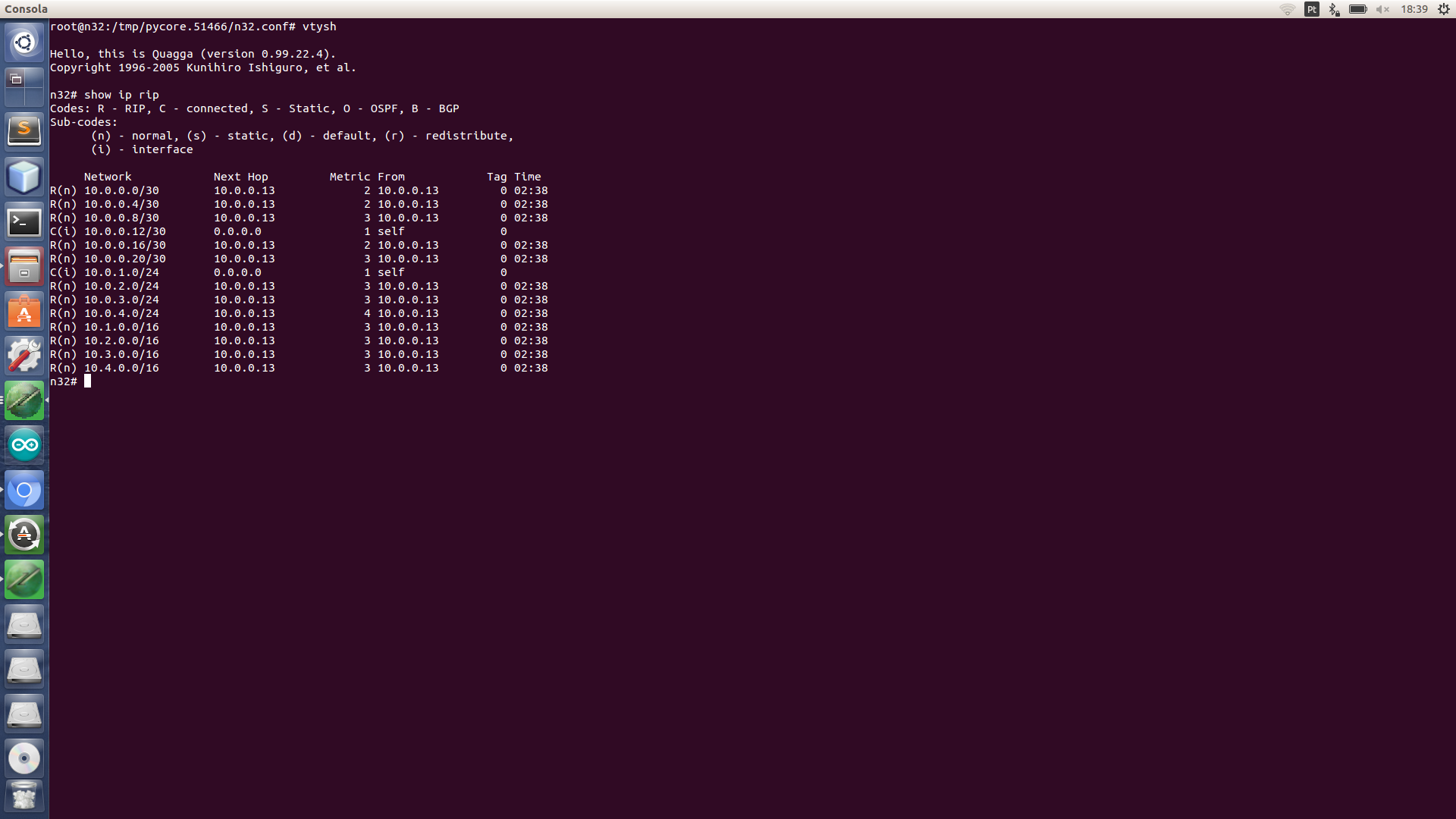


Figura – Show ip rip no router R1.

* O router n32



## Configuração do endereçamento estático

## Configuração das Politicas BGP

A seguinte fase irá abordar por cada sistema autónomo a configuração dos seus *routers* fronteira, contendo uma explicação para cada política implementada, através da especificação dos comandos usados.

### **AS65000**

No enunciado do projeto era referido que este AS deveria ter apenas duas ligações ao exterior: uma ao AS65200 que lhe iria garantir o acesso externo e a conectividade; e uma ligação ao AS65100, com quem o AS6500 mantém uma parceria.

O *router* configurado tem a denominação de “R1”.

Para configurar o *router* o grupo utilizou maioritariamente a sua consola “vtysh”, copiando de seguida o *output* do comando *show running config* para o *zebra*, de modo a garantir que a configuração persistia após o fim da emulação.

Para estabelecer as conexões aos AS vizinhos foi utilizado o comando: “neighbor x.y.z.w remote-as A”, onde “x.y.z.w” se refere ao endereço IP do *router* vizinho e “A” ao número do sistema autónomo desse *router*. O comando foi utilizado duas vezes: uma para ligar ao *router* 192.168.0.13, do AS65100; e outra para ligar ao *router* 192.168.0.25, do AS65200.

Também era pedido no enunciado que o AS6500 fosse implementado como um sistema *stub,* ou seja, um sistema isolado para além da sua saída para o exterior e que não pode servir de trânsito (neste caso o sistema tem duas saídas, mas uma delas é apenas de parceria). Para tal, o grupo decidiu que este AS apenas iria anunciar as suas rotas locais aos seus vizinhos, não anunciando qualquer outra rota que aprendesse. Esta condição foi implementada com recurso a uma *access-list* e um *route-map* (note-se que o *route-map* era desnecessário nesta ocasião, sendo possível utilizar apenas a *access-list*).

A *access-list* criada, de nome “1”, impunha a condição de aceitar apenas rotas locais, rejeitando quaisquer outras. Essa lista, associada aos anúncios que o R1iria enviar para os seus vizinhos, iria filtrar qualquer outra rota que não fosse relativa às suas rotas locais. Para impor a lista aos anúncios de saída o grupo utilizou o comando “neighbor x.y.z.w route-map [NAME] **out**”. Usando este comando para ambos os vizinhos o grupo conseguiu forçar a que o AS6500 se comportasse como um *stub*.

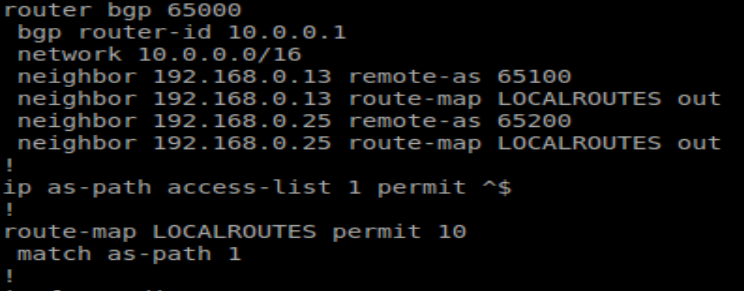


Figura - Show Running Config R1

Na figura 1 podem ser visualizadas as instruções de configuração do R1.

Era ainda pedido no enunciado que a ligação entre o AS65000 e o AS65100 pudesse servir de *backup* para o AS65200, em caso de falhas, havendo, no entanto, preferência pelas ligações diretas. Esta alínea apenas foi implementada nos *routers* R2 e R7, respetivamente do AS65100 e do AS65200, sendo que a sua implementação será documentada de seguida.

No entanto, através da tabela de encaminhamento do R1, pode-se verificar que o AS65000 contém de fato uma ligação *backup* ao AS65200 e que envia tráfego diretamente para o AS65100.

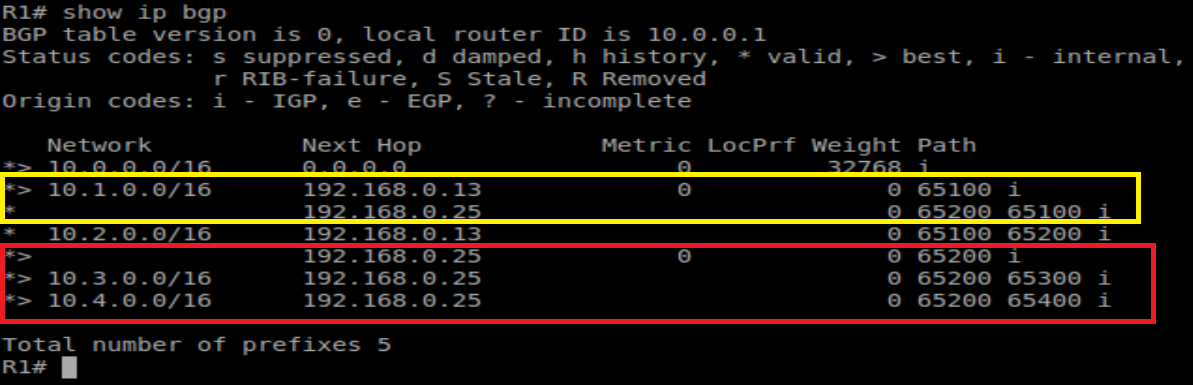


Figura - Show IP BGP R1

Note-se então, na figura 2, as escolhas de encaminhamento do R1.

Para atingir o AS65100 é preferida a ligação direta, como se pode ver no retângulo amarelo. No entanto, para atingir qualquer outro AS o R1 prefere utilizar a ligação com o AS65200, como se pode verificar no retângulo vermelho.

Note-se, na zona entre os dois retângulos, a rota para o AS65200 que passa primeiro pelo AS65100, que não é preferida visto ter um AS\_path maior. Esta rota poderá servir como *backup* em caso de falhas.

Para obter a tabela pretendida do R1 não é necessário que sejam feitas configurações mais avançadas nele próprio.

### AS65100

No enunciado é referido que o AS65100 será um sistema *multihomed*, pelo qual deve, então, conter mais que um *router* fronteira e não permitir servir de trânsito para outros sistemas autónomos. O AS65100 irá, no entanto, servir de trânsito caso a ligação entre o AS65000 e o AS65200 falhe. Para que não sirva de trânsito em mais nenhum caso, o AS65100 não irá divulgar mais que as suas rotas locais, a não ser quando pretende implementar a rota de *backup* entre o AS65200 e o AS65000.

Este AS contém 3 routers BGP: O R2, ligado ao AS65200 e ao AS6500; o R3, ligado ao R5 do AS65300; e, por fim, o R4, ligado ao R6 do AS65300.

Neste AS deverá ainda ser preferido como próximo salto o AS65300 quando o destino é o AS65400. Para finalizar qualquer tráfego destinado ao AS65300 deverá passar pelo R3; e qualquer tráfego destinado ao AS65400 deve passar pelo R4.

**R2**

O *router* R2 tem duas ligações BGP exteriores e duas ligações BGP interiores.

Da mesma forma que para o R1 do AS65000 foi utilizado o comando “neighbor x.y.z.w remote-as A”, no R2 foi utilizado este comando para cada uma das suas quatro ligações BGP. No entanto, para as ligações iBGP foi necessário incluir o comando “neighbor x.y.z.w next-hop-self”, para indicar a esses vizinhos que qualquer rota que o *router* lhes ensinasse teria a si próprio como próximo salto.

Para criar a rota de *backup* foram então criadas duas *acess­-list.*

Uma das listas, denominada de de “1”, iria apenas aceitar rotas locais do R2 e rotas que terminassem no AS65000. A esta lista foi associado um *routemap*, de nome “ROUTESTO65200”, que no entanto não necessitava de ser utilizado. Este mapa foi associado às rotas que o R2 iria anunciar ao *router* R7 (do AS65200), criando assim uma possibilidade de o R7 utilizar o R2 como próximo salto para o R1.

A segunda lista, denominada “diospiro”, tinha o mesmo propósito que a lista “1”, orientada no entanto às rotas que seriam anunciadas a R1. Esta lista apenas permitia que rotas locais ou que rotas terminadas no AS65200 fossem escolhidas. Tendo associado então a lista às rotas anunciadas ao R1, o grupo estabeleceu finalmente que o AS65100 iria servir de trânsito entre os dois AS pretendidos. É também criada a segunda parte da rota de *backup* entre os AS65200 e o AS6500.

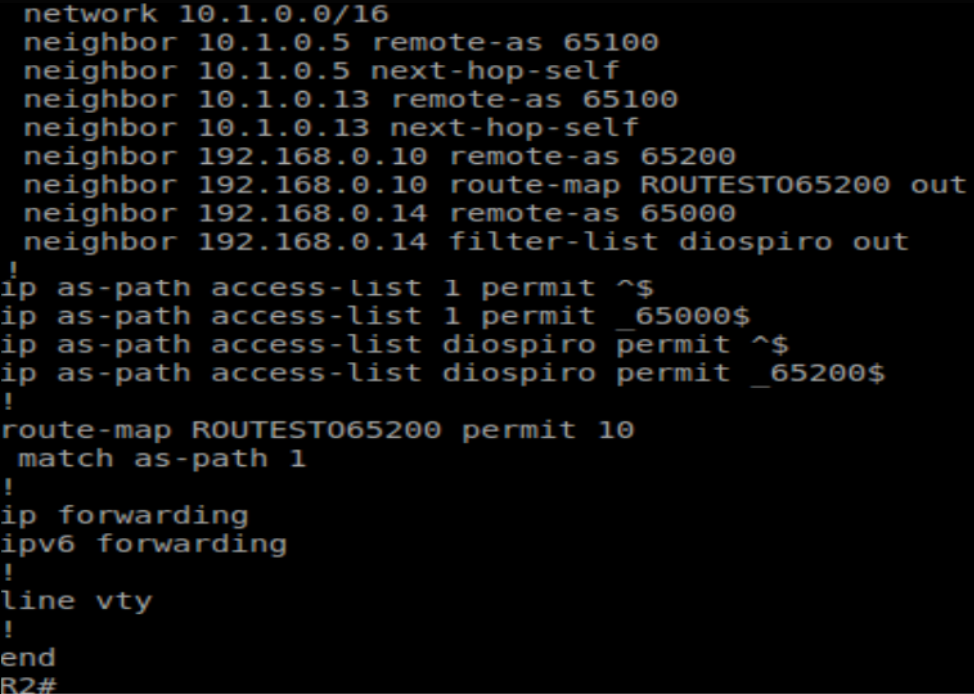


Figura - Show Running Config R2

**R3**

O router R3 está ligado ao R5 do AS65300 e deverá ser o preferido para qualquer trânsito originado no AS65100 e com destino ao AS65300. Para tal, será aumentado o atributo Local\_Pref das rotas com destino ao AS65300 que este o R3 anuncia.

Inicialmente foram estabelecidas as conexões BGP: duas iBGP e uma eBGP. Para tal foram utilizados os comandos já mencionados anteriormente.

Para que o AS65100 não sirva de trânsito, o R3 não anuncia ao seu parceiro eBGP qualquer rota para além das suas redes internas. Para tal foi criada uma *access-list,* denominada “penca” que apenas fazia match com rotas internas ao AS65100. Esta *access-list* serviria de filtro às rotas anunciadas ao R5.

Quando o tráfego tem destino ao AS65400 quer-se que a rota preferida passe pelo AS65300. Então, foram criadas duas *access-list* e um *route-map,* que iria filtrar rotas recebidas do R5,para impor essa condição. Uma das listas, com o nome “broculos” iria apenas aceitar tráfego terminado no AS65400 e ignorar qualquer outro. A outra lista, “couveflor” permitira qualquer rota. Ambas as listas foram então associados a um *route-map* “PREFERE300”, que iria impor duas condições: quando as rotas fizessem *match* com a lista “broculos” seria aumentado o Local\_Pref das rotas recebidas; caso não fosse feito o *match* então não era aumentada a Local\_Pref.

Para que o R3 fosse utilizado como *router* de tráfego para o AS65300 foram criadas outras duas listas e um *route-map* para acompanhar. A *access-list* “lichia” que iria permitir todas as rotas com destino ao AS65300 e negar o resto; e a lista “xuxu” que iria permitir qualquer rota. O *route-map, “*PREFERER3” iria verificar se uma rota fazia *match* com a lista “lichia”: caso fizesse então a Local\_Pref dessas rotas seria aumentada; caso fizesse *match* com a “xuxu” então nada mudava. Este mapa foi então associado às rotas que saíam do R3 com destino tanto ao R2 como ao R4, fazendo com que eles reconhecessem que a melhor rota para o AS65300 passava pelo R3.

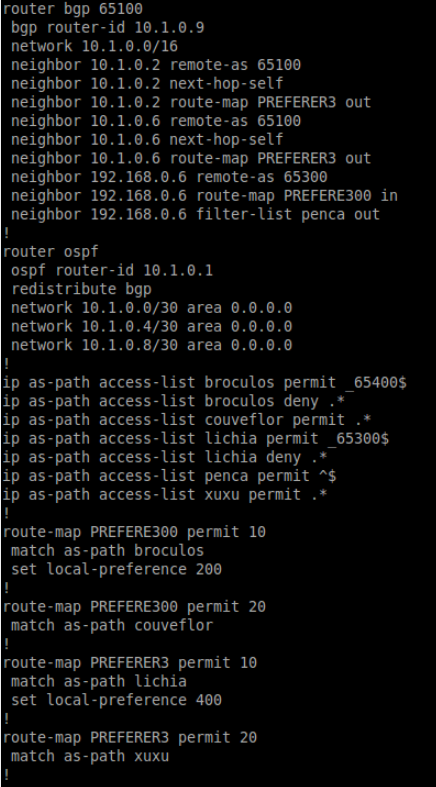


Figura - Show Running Config R3

**R4**

Era pedido que o R4 fosse utilizado para tráfego com destino ao AS65400.

Todas as técnicas utilizadas para configurar o R4 foram utilizadas para configurar o R3: não foram anunciadas rotas para além das locais, através de uma *access-list*; todas as rotas que ele recebia do R6 que terminavam no AS65400 tiveram um aumento do *local­-pref,* utilizando a mesma técnica implementada no R3; e todas as rotas que ele anunciava os seus vizinhos iBGP e que tinham como destino o AS65400 recebiam também um aumento do Local\_Pref, de maneira a que o R4 seja escolhido.

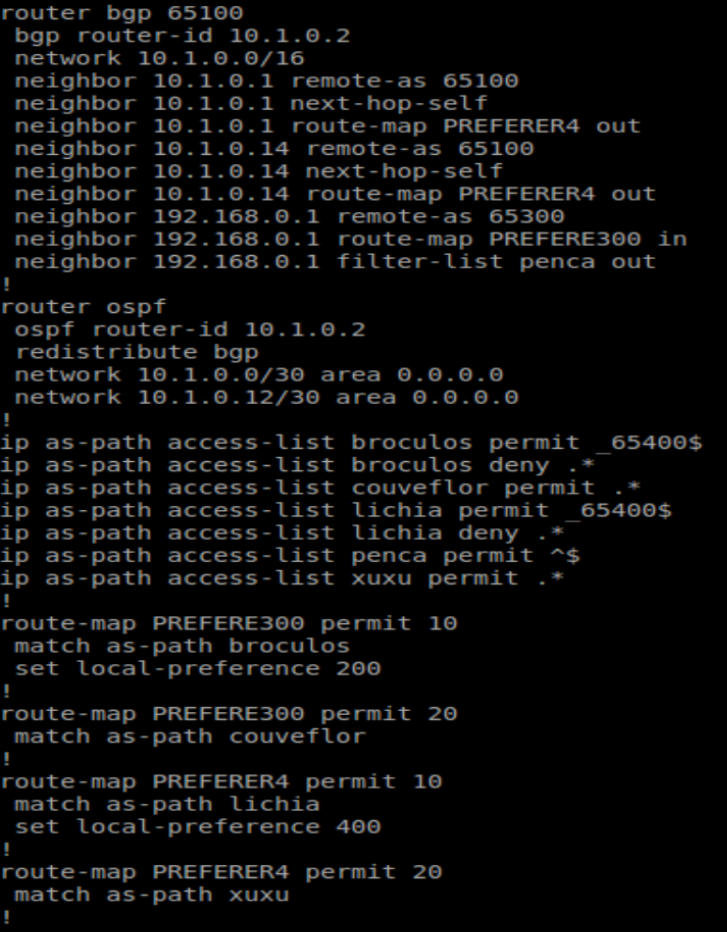


Figura - SHow Running Config R4

Serão de seguida apresentadas as tabelas BGP dos três *routers*.

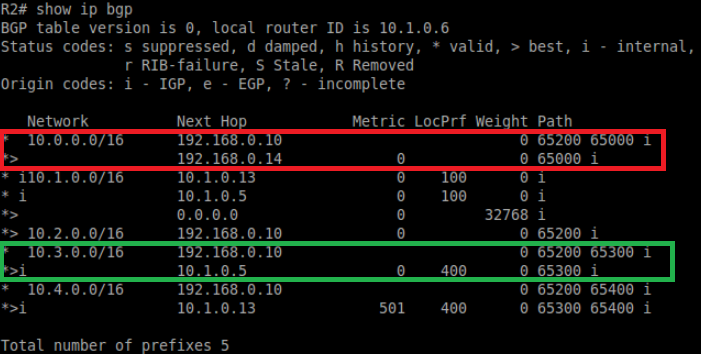


Figura - Show IP BGP R2

Na figura 6 está demonstrada a tabela BGP do R2.

Pode-se verificar que, no retângulo vermelho, para atingir o AS65000 é utilizada a rota direta, existindo no entanto a possibilidade de ir pelo AS65200, o que significa que existe uma rota de *backup*.

No retângulo a verde pode-se verificar as diferentes rotas para atingir o AS65300. Uma delas é interna, e preferida; a outra passa pelo AS65200. Na rota interna verifica-se no entanto que o atributio local preference é maior e que o próximo salto é o endereço 10.1.0.5 (o *router* R3), resultado das configurações feitas de modo a ser escolhido o R3 como *router* para tráfego encaminhado ao AS65300.

Para finalizar o caminho para o AS65400 preferido passa pelo AS65300 primeiro e tem como *next-hop* o *router* 10.1.0.13 (R4). Note-se o valor do campo *“metric”*, que é influenciado pela configuração dos *routers* do AS65300, a ser demonstrado posteriormente.

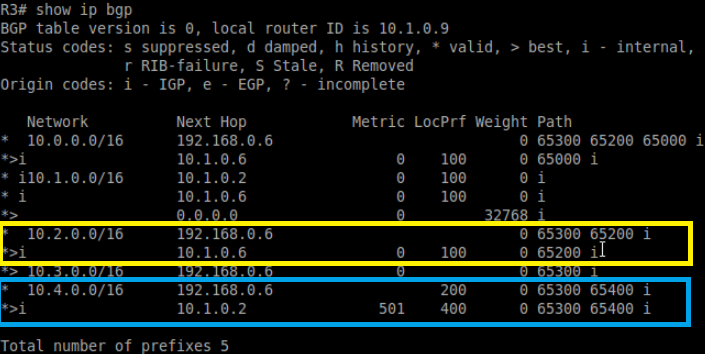


Figura - Show IP BGP R3

Analisando a figura 7 pode-se inferir o conteúdo da tabela BGP do R3.

Tal como seria de esperar o caminho para o AS65000 passa primeiro pelo *router* com o IP 10.1.0.6 (*router* R2), o que seria de esperar, e parte logo do AS. Existe uma outra rota, pelo AS65300, que lhe é anunciado pelo R5. Note-se que não existe uma que passa pelo AS65200, é de assumir que o R2, que lhe deveria passar essa rota, não o faz, tendo em contada que já lhe conta sobre a rota direta.

No retângulo amarelo verifica-se qual o caminho a tomar para o AS65200, é interna e volta a passar pelo R2, mais uma vez, como seria de esperar, tendo em conta ter o menor as\_path.

Entre os retângulos pode-se ver a rota para o AS65300, que é direta. Neste *router* o Local\_Pref é o *default*, visto ser o R3 que anuncia aos outros *routers* esta rota como tendo ma preferência maior

Por fim, no retângulo azul, pode-se ver que o caminho para o AS65400 é uma rota interna e tem como *next hop* o IP 10.1.0.2 (*router* R4). Como o R4 foi configurado para ser utilizado como *router* de trânsito para o AS65400 verifica-se um Local\_Pref maior do que o *default*. Pode-se verificar também que o campo *metric* é diferente do normal, tendo sido alterado pelas configurações dos *routers* do AS65300.

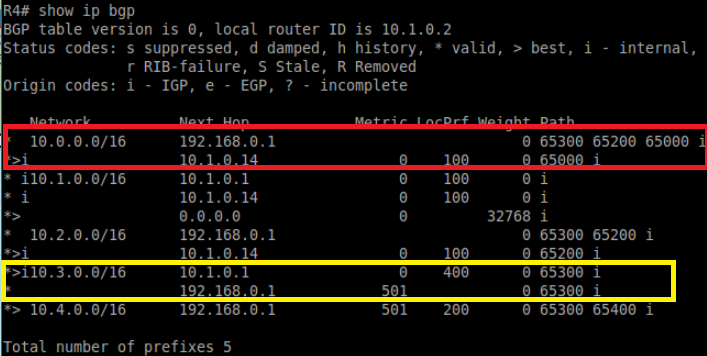


Figura - Show IP BGP R4

Para finalizar têm-se a tabela BGP do *router* R4. Esta tabela é muito semelhante à do R3, há, no entanto algumas diferenças a notar.

No retângulo vermelho verifica-se que a para o AS65000 contia a ser a direta.

Imediatamente acima do retângulo amarelo pode-se verificar que a rota para o AS65200 continua também a ser interna e direta.

Dentro do retângulo amarelo existe uma rota para oAS65300 que passa em primeiro lugar pelo *router* R3, tal como o grupo pretendia com a utilização das *access-lists.* Assume-se que não existem rotas para o AS65300 que passam pelo AS65200 porque o *router* assume que tem já as melhores rotas possíveis.

Para finaliza, abaixo do retângulo amarelo, vê-se que a rota para o AS65400passa em primeiro lugar pela ligação direta que este *router* tem com o R6

### AS65200

O AS65200 serve de ISP para o AS65100 e para o AS6500, devendo aceita e divulgar as suas rotas. Contém um *router*  fronteira, o R7. Este *router* contém uma ligação a 4 *routers* fronteira: com o R2, do AS65100; com o R1, do AS65000; com o R6, do AS65300; e, finalmente, com o R8, do AS65400.

É necessário que o AS65200 mantenha uma rota para o AS65000 que passe pelo ASS65100, para existir como *backup*. O estabelecimento desta rota começou nas configurações dos 2 AS anteriores e irá terminar na configuração deste. Para a implementar era necessário que o AS65200 anunciasse ao AS65100 a rota existente para o AS65000, o que ele faz já automaticamente caso não existam filtros de rotas.

O grupo decidiu não implementar então qualquer filtro no AS65200, o que resultará em o R7 anunciar todas as rotas que conhece a todos os seus vizinhos.

Para configurar as ligações BGP o grupo utilizou o mesmo comando que utilizou nos restantes AS (neighbor x.y.z.w remote-as A).

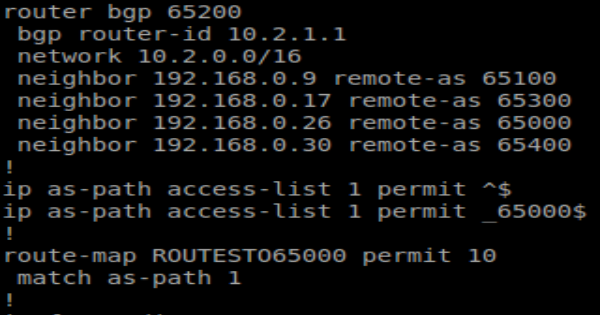


Figura - Show Running Config R7

Na figura 9 pode-se verificar uma *access-list* e um *route-map* que não foram aplicados a qualquer vizinho.

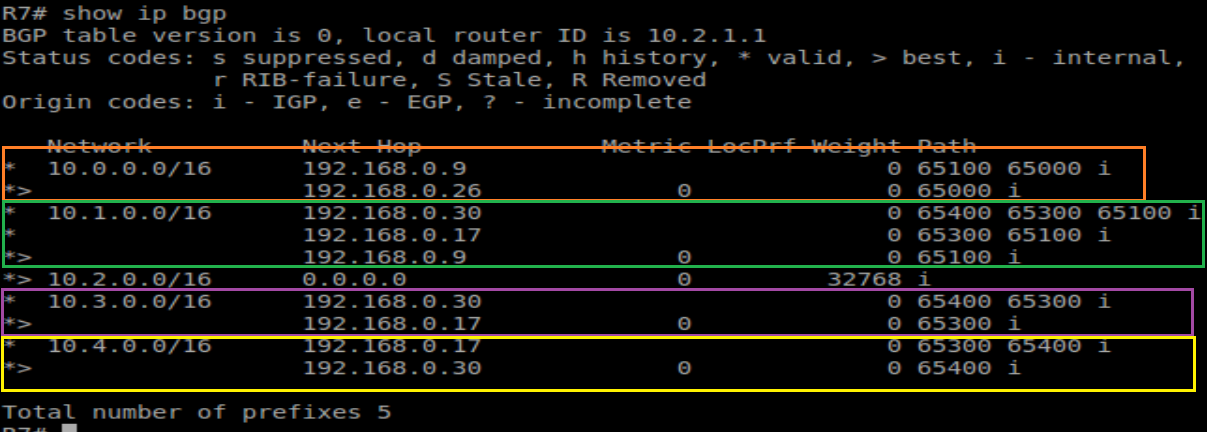


Figura - Show IP BGP R7.

O router R7 recebe todas as rotas possíveis que o R6 e o R8 conhecem (pertencem ao AS65300 e ao AS65400, respetivamente). Do AS65100 recebe as rotas locais e a rota para o AS65000 e deste último recebe apenas as rotas locais.

Tendo em conta o último paragrafo pode-se proceder à analise da figura 10, que mostra a tabela BGP do R7.

Como rota para o AS65000 verificam-se duas entradas, dentro do retângulo laranja, sendo que a rota direta é a preferida, mas pode-se verificar finalmente que a ligação *backup* entre o AS65200 e o AS65000 existe definitivamente.

No retângulo verde verificam-se 3 diferentes rotas para o AS65100: a rota direta, através do R2; uma rota através do AS65400 e do AS65300, como seria de esperar. Passando pelo AS65000 não existe nenhum, visto este apenas anunciar ao R7 as suas rotas locais.

No retângulo roxo verifica-se que a rota preferida para o AS65300 é a rota direta, porque tem menor AS\_path. Tanto à ligação com o AS65300 como com o AS65400 não aplicado qualquer filtro em ambos os sentidos, o que leva o as\_path a ser o fator de decisão.

Por fim, no retângulo amarelo verifica-se que a rota para o AS65400 que é utilizada é a direta.

### AS64300

O AS65300 pretende também ser ISP do AS65100 e, assim, deverá aceitar todas as suas rotas e anunciá-las. Qualquer tráfego este AS receba com destino ao AS65100 deve também ser reencaminhado pelo *router* R5, o que será feito com recurso a *access-lists* e a *rout-maps*.

O AS65300 não irá anunciar ao AS65400 às rotas que conhece para o AS65000, o que será implementado com recurso a um *access-list*.

Este AS contém dois *routers* fronteira: o R5, que tem uma ligação eBGP com o R3 do AS65100; e o R6, que contém uma ligação com os *routers* R4, R7 e R8 dos AS 65100, 65200 e 65400, respetivamente. O R5 e o R6 mantém uma relação iBGP entre si.

Para finalizar o AS65300 pretende influenciar o AS65100 de modo a fazer com que este último faça passar o tráfego que é dirigido a si próprio pelo R5. Pretende também que qualquer tráfego com destino ao AS65100 passe de também pelo *router* R5.

A configuração das parcerias BGP, tanto externas como internas, foram feitas através do comando “neighbor x.y.z.w remote-as A” e “neighbor x.y.z.w next-hop-self” para as parcerias iBGP.

**R5**

Pretendia-se que o R5 servisse como *router* de trânsito para o AS65100, para tal implementaram-se duas *access-list* e um *route-map*, duma maneira análoga ao que foi feito no R4 e no R3. Uma das listas, de nome “papaia”, iria permitir todas as rotas terminadas no AS65100 e rejeitar as restantes. A outra lista, denominada “mamao”, iria aceitar todas as restantes rotas. O *route-map* que utiliza ambas as listas iria primeiro verificar se alguma rota fazia *match* com a lista “papaia” e caso fizesse iria aumentar o atributo Local\_Pref, de modo a que estas rotas fossem preferidas; caso não fizesse *match*, então não seria feita qualquer mudança. Este mapa serviu para filtras as rotas que eram anunciadas ao seu vizinho iBGP, o *router* R6. Deste modo foi influenciada a escolha da rota a tomar para o AS65100.

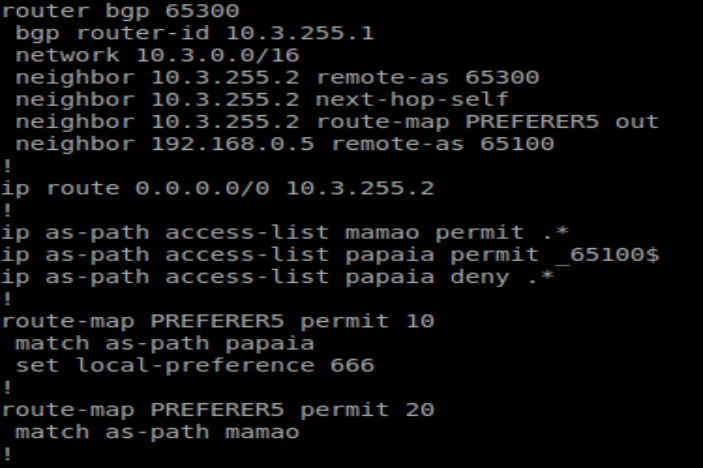


Figura - Show Running Config R5

**R6**

Para não anunciar ao AS65400 as rotas para o AS65000 foi implementada no R6 uma *access-list,* de nome “1” que negava rotas terminadas no AS65000, mas permitia rotas terminadas nos restantes AS. Esta lista foi utilizada num *route-map*, de nome “DENYROUTESTO65000” que é na verdade desnecessário. O mapa foi anexado às rotas que seriam anunciadas ao AS65400, filtrando assim qualquer rota com fim no AS65000.

Para tentar influenciar o AS65100 a escolher este *router* como ponto de tráfego para o AS65300 e para o AS65400 foram implementadas mais duas *acess-list* e um *route-map*. A *acess-list* “jaca” iria aceitar rotas terminadas no AS65400 ou nas rotas locais do AS65300. Uma segunda lista “marmelo” iria já aceitas todas as restantes rotas. O *route-map* “INFLUENCE65100”, anexado às rotas anunciadas ao AS65100, ao verificar que uma rota fazia *match* com a lista “jaca” iria mudar o atributo “*metric”* para um valor elevador. Note-se que o valor o campo Multi Exit Discriminator (MED) tenta influencia a escolha de rotas dos AS a quem estas são anunciadas. Um MED mais alto iguala a menor preferência por uma rota. Então, para para fazer com que o R5 fosse considerado para lidar com todo o trâfego aumentou-se o MED as rotas anunciadas pelo R6.

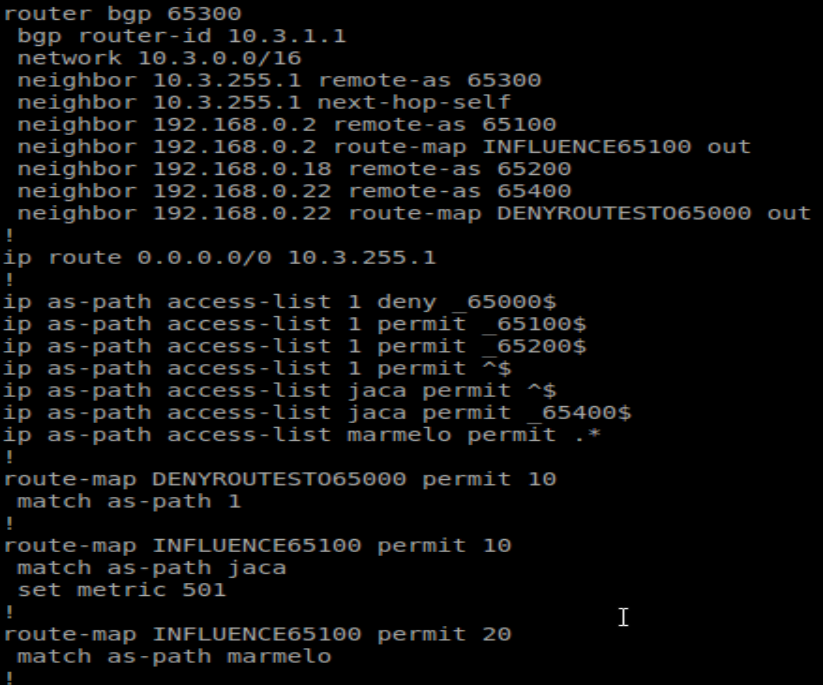


Figura - Show Running Config R6.

De seguida poder-se-á visualizar as tabelas BGP de ambos de tanto o R5 como o R6.

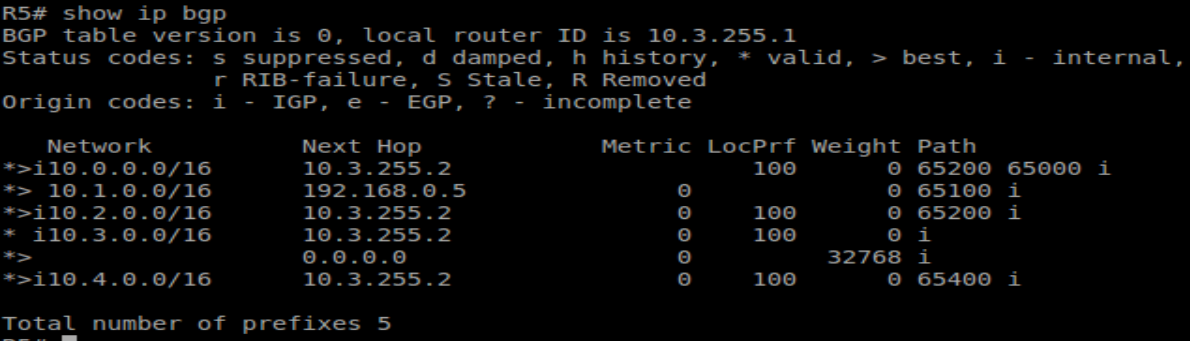


Figura - Show IP BGP R5.

Na figura 13 pode-se verificar que a tabela do R5 é curta. Tenha-se em conta que o AS65100 não anuncia aos *routers* do AS65300 qualquer rota para além das locais.

Para o AS65000 apenas conhece a rota que lhe é anunciada pelo seu vizinho R6, e mais nenhuma, visto o AS65100 não a anunciar.

Com destino ao AS65100 o R5 apenas demonstra a melhor rota possível, aquela que passa pela sua ligação direta. O R6 não lhe anuncia outra rota, visto que a própria rota que ele recebe do R5 para o AS65100 é tida como a melhor.

Para o AS65200 e para o AS65400 ele conhece apenas a rota que o R6 lhe anuncia, visto que de qualquer forma as rotas possíveis para o AS65200, e também para o AS65400, irão passar pelo R6.

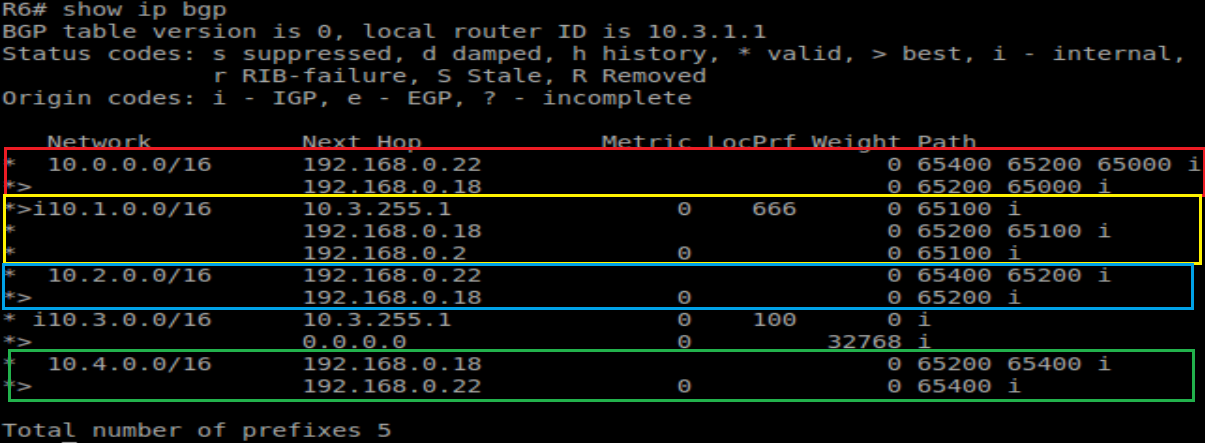


Figura - Show IP BGP R6

O R6 já demonstra bastantes mais rotas, como se pode verificar na figura 14. Este *router* recebe diretamente anúncios de rotas tanto do AS65400 como do AS65200, para não mencionar as rotas locais do AS65100. Seria de esperar então um maior número de rotas, tendo em conta que o AS65400 é um sistema de trânsito, assim como o AS65200.

No retângulo vermelho pode-se verificar as possíveis rotas para o AS65000, visto que o AS65100 não anuncia a sua. É escolhida aquela com o AS\_path mais curto.

Em baixo, no retângulo amarelo, verificam-se 3 rotas possíveis para o AS65100. A rota direta, e preferida, que é também interna; a rota que passa pelo AS65200; e a rota que passa pelo AS65400. Como já foi mencionado no primeiro parágrafo do comentário às rotas do R6, este número de rotas é esperado.

No retângulo azul verifica-se que o caminho escolhido para o AS65200 é o direto. No verde verifica-se também que o caminho direto para o AS65400 é o preferido.

### AS65400

Para finalizar a análise das implementações das políticas de encaminhamento vai-se analisar o AS65400. Este AS apenas pretende obedecer a uma política, a de preferir o AS65300, em detrimento do AS65200, para tráfego com destino ao AS65100.

Este AS contém um *router* fronteira, o R8, que estabelece duas parcerias eBGP: uma com o R6, do AS65300 e outra com o R7, do AS65200.

Para implementar então a preferência pelo AS65300 a caminho para o AS65100 foram utilizadas duas *access-list*  e um *route-map.* A primeira *access-list*, de nome “chocolate”, iria permitir qualquer tráfego com destino ao AS65100 e recusar o resto. A segunda lista, denominada “bolotas” iria permitir qualquer tráfego. O mapa “PREFERE300” iria verificar que rotas faziam *match* com a lista “chocolate” e aumentar o Local\_Pref dessas rotas. Caso não houvesse *match* o mapa não faria nada. Este mapa serviu para filtrar as rotas de entrada enviadas pelo *router* R6. Sendo assim, qualquer rota recebida pelo R6, do AS65300, e com destino ao AS65100 tinha um Local\_Pref maior, sendo escolhida como rota preferida.

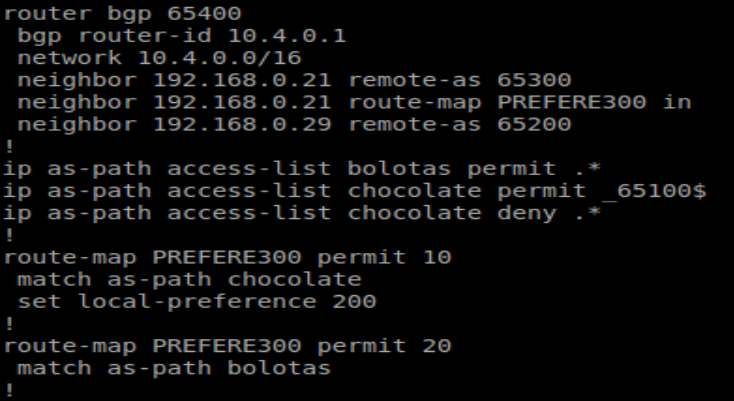


Figura - Show Running Config R8

De seguida poder-se-á visualizar a tabela BGP do R8.

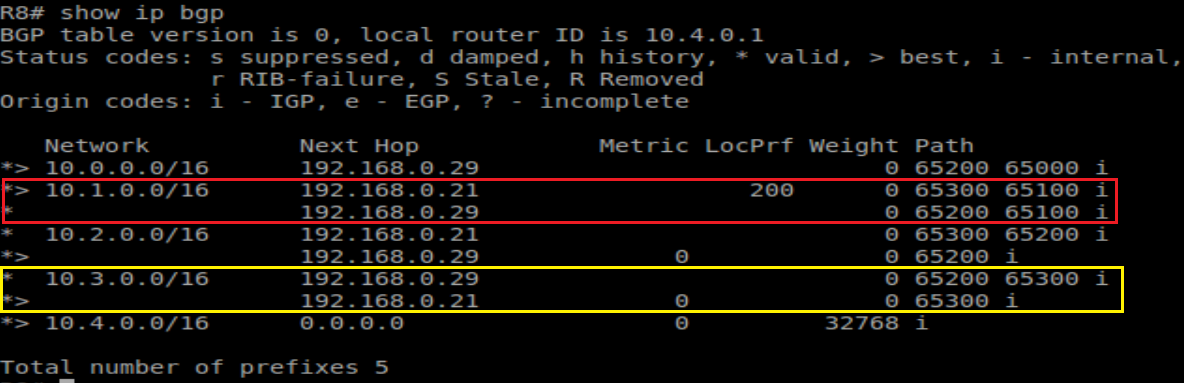


Figura - Show IP BGP R8

Na figura 16 pode-se verificar que a tabela BGP do R8 não é muito extensa, mesmo tendo em conta que este recebe todas as rotas que o R7 conhece. Não recebe, por exemplo, a rota para o AS65000 que passa pelo AS65300, como já foi mencionado anteriormente.

Então, para atingir o AS65000 o R8 apenas conhece a alternativa do AS65200.

Por sua vez, para atingir o AS65100 já existem 2 rotas. Note-se que a rota que passa pelo AS65300 é a preferida devido ao atributo Local\_Pref maior, tal como se pretendia.

Entre os retângulos verifique-se que para atingir o AS65200 exitem as rotas pelo AS65300 e a direta. Sendo que a direta é a favorita, visto o seu as\_path menor e não havendo políticas que influenciem a escolha.

Dentro do retângulo amarelo pode-se ver que existem duas rotas para o AS65300, da mesma maneira que existem duas rotas para AS65200. É escolhida aquela com menor as\_path.

### Conclusão da Análise das Tabelas

Em grande parte verifica-se que as rotas que se pretendia utilizar são realmente aquelas que aparecem nas tabelas BGP dos *routers*. O grupo conseguiu implementar com sucesso as diferentes po