1. Na declaração match deve ser colocado uma prefix-list uma vez que permite numa única declaração fazer o match com todas as subnets de 176.16.0.0/16 com menos de 24 bits, na declaração set deve ser colocado um valor de local preference superior a 100 (valor por omissão) de modo a que as rotas afetadas fiquem com uma local preference superior a por omissão e logo mais preferida.
   Por fim o route map deve ser aplicado em IN ao vizinho 10.1.1.2 de modo a alterar a local preference nas rotas anunciadas por este.

   Comandos do route map (no teste não seria necessário colocar os comandos):

   Route map localpref permit 10
   match ip address prefix-list Myslist
   set local preference 150

   onde a Mylist seria:
   ip prefix-list Mylist permit 176.16.0.0/16 le 24.

2. a. De modo a ter a maior conectividade possível as vizinhanças devem ser feitas usando as interfaces de loopback dos vizinhos, dessa forma desde que exista conectividade para essas interfaces a vizinhança está ativa. Para isto ser possível é necessário que se conheçam caminhos para essas interfaces, o que neste cenário acontece pois todas as redes internas estão incluídas no OSPF. Assim as vizinhanças devem ser feitas para os endereços 10.2.2.2. e 10.1.1.1, deve ser feito também o comando update-source com o loopback de A (10.3.3.3) de modo a que na vizinhança o endereço de origem dos pacotes seja também o loopback de A. Finalmente neste caso não é necessário o next-hop-self uma vez que o next-hop das rotas eBGP vai ser uma das redes 192.168.30.4/30 ou 192.168.30.0/30 e estas estão no OSPF pelo que os vizinhos iBGP as conseguem alcançar.

   b. Neste caso as vizinhanças internas teriam de ser feitas por A usando as interfaces físicas 10.1.20.1 e 10.1.30.1 uma vez que não seriam conhecidos caminhos para os loopbacks. Do mesmo modo passaria a ser necessário o next-hop-self uma vez que as redes 192.168.30.4/30 ou 192.168.30.0/30 deixam de ser conhecidas nos vizinhos que assim não conseguiriam chegar ao next-hop das rotas BGP. Finalmente e como o next-hop self substitui o next-hop pelo endereço do loopback de A seria necessário criar uma rota estática para a rede 10.3.3.0/24.

3. A BEB vai colocar o cabeçalho 802.1ah ou seja encapsula a frame internet original com um novo cabeçalho MAC com o formato 802.1ah. Esse cabeçalho tem os campos: source address (B-MAC SA) com o endereço MAC da BEB; destination address (B-MAC DA) com o endereço de grupo multicast da Backbone VLAN 20, ; I-SID 150 e B-VLAN 20. Uma vez que é um unknown unicast a frame tem de ir para todas as BEB da B-VLAN 20. A BEB guarda o mapeamento do MAC de origem do Cliente 00:00:00:20 com o porto 4 e a VLAN de cliente de modo a saber que quando recebe uma frame com esse endereço de destino de cliente no I-SID 150 da B-VLAN 20 a deve enviar para a porta 4.

4. Numa solução SD-WAN vai existir um terceiro componente para além dos da figura. Esse componente é um controlador SDN que vai controlar a configuração e/ou o funcionamento do data plane (dependendo do tipo de SDN) dos CPEs de modo a que estes utilizem um dos dois caminhos possíveis entre eles. Terá de haver um túnel de comunicação de controlo entre o controlador e os CPEs bem como túneis de dados estabelecidos entre os CPEs. O túnel de dados através da internet pode ser um túnel IPsec por exemplo e o túnel através de MPLS ser um circuito fornecido por Service Provider como um MPLS pseudowire ethernet. O controlador pode depois dinamicamente alterar a forma como os CPEs usam os dois tipos de transporte.

5. Os campos relevantes da mensagem FlowMod para obter o comportamento pretendido são:
   Match com os campos: Ethertype=IPv4; IpProtocol=TCP; SourceMAC= 01:23:45:67:89:AB;
   IntructionList = Apply Actions;
   ActionsList = PushVLAN 20; Output port 6;
   HardTimeout = 20.

6. Deve ser enviada proativamente uma mensagem FlowMod para instalar uma regra com o match IP destino=192.168.16.1 , ethertype = IPv4 ; IpProto = TCP; transport port 25 e com a action forward controller. Essa regra faz com que o primeiro pacote de uma nova ligação para o porto 25 com esse IP de destino seja recebido no controlador numa mensagem Packet_IN. Ao receber esse pacote o controlador envia uma mensagem FlowMod para instalar uma regra com um Match semelhante ao anterior mas agora também com o IP de origem e o porto de transporte de origem de modo a que só os pacotes desta ligação específica façam match. Este FLowMod tem também uma prioridade superior e um Idle_timeout. Assim, durante a ligação os pacotes passam a fazer match com esta nova regra, quando a ligação terminar e após o idle_timeout é enviada uma mensagem Flow_Removed ao controlador que pode obter a duração nas estatísticas do Flow_Removed.

7. O VIM é responsável por controlar e gerir os recursos físicos necessários para executar e interligar as funções virtuais de rede (VNFs) necessárias a um serviço. Desse modo, é responsável por criar as VMs ou Containers onde as VNFs vão ser executadas bem como interligar pela rede esses recursos dentro da sua esfera de controlo. O SDN vai ser importante para a virtualização da parte de rede, ou seja para controlar a interligação das VMs e Containers criados de forma rápida e flexível.