

RouterOS 5 MPLS – polecenia wykorzystywane podczas laboratorium

1. Stanowisko laboratoryjne

Na każdym komputerze stworzone są 4 maszyny wirtualne:

- Stanowisko nieparzyste: R1, R2, R3
- Stanowisko parzyste: R4, R5, R6

Interfejs “VMware Network Adapter VMnet9” systemu Windows (stanowiska fizycznego) jest podłączony do wirtualnego interfejsu routera:

- Stanowisko nieparzyste – R1,
- Stanowisko parzyste – R5.

Wirtualne routery R2 i R3 połączone są za pośrednictwem fizycznego łącza między interfejsami eth2 stanowisk fizycznych.

2. Układ poleceń i mechanizm pomocy

Polecenia pogrupowane są w grupy tematyczne, tworzące strukturę drzewiastą. Np.:

- grupa ip – dotyczy konfiguracji protokołu IPv4 i z kolei zawiera grupy, np.:
 - address – grupa umożliwiająca konfigurację adresów IPv4,
 - route – grupa umożliwiająca konfigurację tras routingu IPv4.
- grupa routing – dotyczy konfiguracji protokołów routingu i z kolei zawiera grupy, np.:
 - rip – konfiguracja protokołu RIP,
 - ospf – konfiguracja protokołu OSPFv2 (IPv4),
 - ospfv3 – konfiguracja protokołu OSPFv3 (IPv6).
- etc.

Jeśli z linii poleceń wpisujemy nazwę grupy:

```
[admin@r1] > ip address
```

to aktualna grupa (coś w rodzaju aktualnego folderu w przypadku systemu plików) zmieni się odpowiednio, co widać po wyświetlanym znaku zachęty:

```
[admin@r1] /ip address >
```

W tym momencie naszą aktualną grupą to ip address.

Aby wrócić poziom wyżej należy użyć polecenia „..”

```
[admin@r1] /ip address > ..
```

```
[admin@r1] /ip >
```

Aby wrócić do korzenia drzewa grup, należy użyć polecenia „/”.

```
[admin@r1] /ip address > /
```

```
[admin@r1] >
```

Grupy mogą zawierać podgrupy (jak widać powyżej) oraz polecenia.

Do przeglądania zawartości grup należy użyć polecenia „?” – wpisane od nowej linii wypisze ono całą zawartość danej grupy (podgrupy i polecenia). Podgrupy zaznaczone będą kolorem niebieskim, a polecenia fioletowym.

Aby wydać jakieś polecenie, należy podać je, poprzedzone prowadzącą do niego ścieżką. Na przykład, aby dodać nowy adres podajemy polecenie:

```
[admin@r1] > ip address add address=192.168.1.1/24 interface=ether1
```

jeśli naszą aktualną grupą jest korzeń drzewa grup. Jeśli zmienimy naszą aktualną grupę, zmieni się też polecenie które musimy wpisać:

```
[admin@r1] > ip address
```

```
[admin@r1] /ip address > add address=192.168.1.1/24 interface=ether1
```

Polecenie “?” służy nam nie tylko do wyświetlania zawartości grup. Można go użyć również do wyświetlania pomocy dotyczącej funkcji i parametrów konkretnego polecenia. Np.:

```
[admin@r1] > ip address add ?
```

Powoduje wyświetlenie funkcji i listy parametrów polecenia (parametry wyświetlane są w kolorze zielonym).

```
[admin@r1] > ip address add address=?
```

Powoduje wyświetlenie informacji na temat możliwych wartości parametru *address* podanego polecenia.

W celu automatycznego dopełniania poleceń, należy użyć przycisku TAB. Np.: jeśli naciśniemy TAB wpisawszy polecenie:

```
[admin@r1] > ip ad
```

to zostanie ono uzupełnione do:

```
[admin@r1] > ip address
```

gdyż z grupy *ip* tylko polecenie „*address*” zaczyna się na „*ad*” – czyli można je było jednoznacznie dopełnić. Jeśli polecenia nie można jednoznacznie dopełnić, pojedyncze naciśnięcie TAB nie przyniesie efektu. Można jednak w takim przypadku nacisnąć TAB dwa razy, co spowoduje wypisanie wszystkich pasujących możliwości. Np.:

```
[admin@r1] > ip a
```

```
accounting address arp
```

Jeśli dane polecenie może zostać jednoznacznie uzupełnione z użyciem TAB, to może pozostać w takiej skróconej formie i zostanie wykonane poprawnie, np.:

```
[admin@r1] > ip ad a a=192.168.1.1/24 i=ether1
```

da ten sam efekt, co:

```
[admin@r1] > ip address add address=192.168.1.1/24 interface=ether1
```

3. Składnia poleceń

UWAGA: W poniższej instrukcji wszystkie polecenia podane są z „pełną ścieżką”, tzn. w postaci, w której należy je podać, jeśli znajdujemy się u korzenia drzewa grup.

3.1. Interfejsy:

interface print – wypisanie wszystkich interfejsów

interface ethernet print – wypisanie wszystkich interfejsów Ethernet

interface ethernet disable/enable <NAZWA> - wyłączenie (disable) lub włączenie (enable) interfejsów Ethernet o podanej nazwie. Zamiast nazw można też użyć numerów interfejsów.

Przykład:

```
interface ethernet disable 1,3
```

```
interface ethernet enable ether1,ether5
```

interface bridge print – wypisanie wszystkich interfejsów typu most

interface bridge add name=<NAZWA> auto-mac=no admin-mac=<MAC> - utworzenie nowego interfejsu typu most o zadanym adresie MAC. Wybrane adresy MAC powinny być unikalne w tworzonym systemie sieciowym,

Przykład:

```
interface bridge add name=bridge1 auto-mac=no admin-mac=00:00:00:00:00:01
```

interface bridge port add interface=<INTERFEJS> bridge=<NAZWA> – dodanie interfejsu do mostu o określonej nazwie,

Przykład:

```
interface bridge port add interface=ether1 bridge=bridge1
```

interface bridge remove <NAZWA> – usunięcie mostu o określonej nazwie.

Przykład:

```
interface bridge remove bridge1
```

3.2. Adresy IP

ip address print – wypisanie wszystkich nadanych adresów IP,

ip address add address=<ADRES/MASKA> interface=<INTERFEJS> – przypisanie adresu do interfejsu o podanej nazwie,

Przykład:

```
ip address add address=192.168.2.2/24 interface=ether1
```

ip address remove numbers=<NUMER(Y)> – usunięcie adresu (adresów) IP z interfejsów o podanych numerach (patrz *ip address print*).

Przykład:

```
ip address remove numbers=0,1
```

3.3. Routing statyczny

ip route print – wypisanie wszystkich (statycznych i dynamicznych) tras znanych na danym routerze,

ip route add dst-address=<ADRES_SIECI/MASKA> gateway=<ADRES lub INTERFEJS> - dodanie trasy statycznej prowadzącej do podanej sieci, nakazującej wysyłać ruch przez podany interfejs (dostarczanie lokalne – odbiorca jest bezpośrednio podłączony do tego interfejsu) lub na podany adres IP kolejnego routera na trasie (dostarczanie zdalne),

Przykład:

```
ip route add dst-address=192.168.80.0/24 gateway=10.10.1.1
```

ip route remove numbers=<NUMER(Y)> - usunięcie tras o podanych numerach (do odczytania z *ip route print*).

3.4. RIP

routing rip print – wyświetla ogólne informacje na temat konfiguracji protokołu RIP (np. wartości timerów),

routing rip interface print – wyświetla listę interfejsów, których dotyczy szczegółowa konfiguracja protokołu RIP (patrz polecenie *routing rip interface add*),

routing rip route print – wyświetla listę tras routingu uzyskanych z użyciem RIP,

routing rip set update-timer=<CZAS> – ustawia częstość rozsyłania rozgłoszeń RIP na podaną wartość,

Przykład:

```
routing rip set update-timer=5s
```

routing rip set timeout-timer=<CZAS> - ustawia czas, po którym nie odświeżona trasa zostaje uznana za nieaktualną i przestaje być używana (ale pozostaje w tablicy routingu),

Przykład:

```
routing rip set timeout-timer=1m
```

routing rip set garbage-timer=<CZAS> - ustawia czas, po którym nieodświeżana trasa uznana za nieaktualną zostaje usunięta z tablicy routingu.

routing rip set redistribute-connected=yes|no – włączenie (yes) i wyłączenie (no) rozgłaszania przez protokół RIP informacji o wszystkich sieciach dołączonych bezpośrednio do routera. W przypadku wyłączenia tej opcji (domyślnie), rozgłaszane są informacje wyłącznie o sieciach z listy sieci obsługiwanych przez protokół RIP, tworzonej poleceniem *routing rip network add*.

routing rip network add network=<ADRES_SIECI/MASKA> - powoduje dodanie do listy sieci obsługiwanych przez protokół RIP sieci o danym adresie. Wiadomości RIP są rozgłaszane i odbierane tylko przez interfejsy posiadające adresy IP należące do sieci z tej listy. Ponadto protokół rozsyła wyłącznie informacje na temat sieci z danej listy, chyba, że użyjemy opcji *redistribute-...*

Przykład:

```
routing rip network add network=192.168.80.0/24
```

routing rip network remove numbers=<NUMER(Y)> - powoduje usunięcie z listy tworzonej poprzednim poleceniem wpisów o określonych numerach (wyświetlenie listy – *routing rip network print*).

Przykład:

```
routing rip network remove numbers=0,1
```

routing rip interface add interface=<INTERFEJS> passive=yes - dodanie nowego wpisu do listy szczegółowej konfiguracji interfejsów na potrzeby protokołu RIP. Wpis ten powoduje, iż dany interfejs będzie działał w trybie pasywnym, tzn. nie będą przez niego rozsyłane (lecz dalej mogą

być odbierane) wiadomości RIP. Brak wpisu dotyczącego określonego interfejsu oznacza, iż ma on działać wg konfiguracji domyślnej (czyli, między innymi, w trybie aktywnym).

Przykład:

```
routing rip interface add interface=ether2 passive=yes
```

routing rip interface remove numbers=<NUMER(Y)> - powoduje usunięcie z listy tworzonej poprzednim poleceniem wpisów o określonych numerach (wyświetlenie listy – *routing rip interface print*). Brak wpisu dotyczącego określonego interfejsu oznacza, iż ma on działać wg konfiguracji domyślnej.

3.5. OSPF

routing ospf instance print – wypisanie informacji o instancjach OSPF,

routing ospf area print – wypisanie informacji o strefach OSPF,

routing ospf network print – wypisanie wszystkich sieci dodanych do stref OSPF,

routing ospf interface print – wypisanie wszystkich interfejsów przez które rozgłaszane są komunikaty,

routing ospf neighbor print – wypisanie wszystkich wykrytych sąsiadów OSPF,

routing ospf route print – wypisanie tras routingu uzyskanych z użyciem OSPF,

routing ospf instance add name=<NAZWA> - tworzy nową instancję protokołu OSPF pod podaną nazwą.

routing ospf instance remove <NAZWA> - usuwa instancję protokołu OSPF o podanej nazwie.

routing ospf instance disable|enable <NAZWA> - wyłączenie (disable) lub włączenie (enable) instancji o podanej nazwie. Zamiast nazwy można też użyć numerów instancji (do odczytania z *routing ospf instance print*).

Przykład:

```
routing ospf instance disable default
```

```
routing ospf instance enable 0
```

routing ospf area add area-id=<IDENTYFIKATOR> name=<NAZWA> - tworzy nową strefę OSPF o podanej nazwie oraz identyfikatorze. Identyfikator ma postać analogiczną do adresu IP.

Przykład:

```
routing ospf area add area-id=0.0.0.0 name=backbone
```

routing ospf area remove <NAZWA> - usuwa strefę OSPF o podanej nazwie.

routing ospf network add network=<ADRES_SIECI/MASKA> area=<NAZWA_STREFY> – dodanie sieci IP do strefy OSPF. Spowoduje to włączenie rozgłaszania i odbieranie wiadomości OSPF przez interfejsy, które posiadają przypisany adres IP należący do tej sieci. Ponadto protokół rozsyła wyłącznie informacje na temat sieci z danej listy, chyba, że użyjemy opcji *redistribute-...*

Przykład:

```
routing ospf network add network=10.10.1.0/24 area=backbone
```

Wskazanie instancji OSPF, że sieć 10.10.1.0/24 należy do strefy OSPF o nazwie backbone.

routing ospf network remove numbers=<NUMER(Y)> - usunięcie wpisów z listy sieci dodanych do stref OSPF (wyświetlenie listy - *routing ospf network print*).

Przykład:

```
routing ospf network remove numbers=1,5
```

routing ospf instance set redistribute-connected=ad-type-1 <NAZWA> – włącza rozgłaszanie przez protokół OSPF informacji o wszystkich sieciach, do których router jest bezpośrednio podłączony (będą one rozgłaszane jako tak zwane sieci external – zewnętrzne w stosunku do strefy OSPF). Jako NAZWA podajemy nazwę instancji OSPF

routing ospf instance set redistribute-connected=no <NAZWA> – wyłącza rozgłaszanie przez protokół OSPF informacji o wszystkich sieciach do których router jest bezpośrednio podłączony. Jako NAZWA podajemy nazwę instancji OSPF

routing ospf interface add interface=<NAZWA> passive=yes|no – dodanie nowego wpisu do listy szczegółowej konfiguracji interfejsów na potrzeby protokołu OSPF. Wpis ten powoduje, iż dany interfejs będzie działał w trybie pasywnym, tzn. nie będą przez niego rozsyłane (lecz dalej mogą być odbierane) komunikaty OSPF. Brak wpisu dotyczącego określonego interfejsu oznacza,

iż ma on działać wg konfiguracji domyślnej (czyli, między innymi, w trybie aktywnym).

routing ospf interface remove numbers=<NUMER(Y)> - powoduje usunięcie z listy tworzonej poprzednim poleceniem wpisów o określonych numerach (wyświetlenie listy – *routing ospf interface print*). Brak wpisu dotyczącego określonego interfejsu oznacza, iż ma on działać wg konfiguracji domyślnej.

routing ospf area range print – wyświetla listę przedziałów, do których przeprowadzane jest sumowanie sieci IP (patrz **Błąd! Nie można odnaleźć źródła odwołania.**) przy rozsyłaniu informacji o trasach poza granice danej strefy.

routing ospf area range add area=<NAZWA> range=<ADRES_SIECI/MASKA> - powoduje dodanie do strefy o określonej nazwie nowego przedziału sumowania (patrz **Błąd! Nie można odnaleźć źródła odwołania.**).

routing ospf area range remove <NUMER(Y)> - powoduje likwidację przedziału sumowania o podanym numerze (do odczytania z listy *routing ospf area range print*).

Rozszerzenie OSPF dla potrzeb MPLS TE:

routing ospf instance set mpls-te-area=<STREFA> mpls-te-router-id=<INTERFEJS> <NAZWA_INSTANCJI> – włączenie w podanej strefie i instancji OSPF rozszerzenia pozwalającego na transportowanie informacji potrzebnych do działania mechanizmu MPLS TE.

Przykład:

```
routing ospf instance set mpls-te-area=0.0.0.0 mpls-te-router-id=bridge1 default
```

3.6. MPLS

mpls local-bindings print – wylistowanie lokalnie stworzonych znaczników,

mpls remote-bindings print – wylistowanie znaczników otrzymanych od innych routerów,

mpls forwarding-table print – wylistowanie tablicy odwzorowań znaczników przychodzących na wychodzące,

mpls set dynamic-label-range=<RANGE> – ustawienie przedziału, w którym mają się zawierać wartości tworzonych znaczników MPLS. Uwaga: przedział zawierać musi co najmniej 1024 wartości i zaczynać się od wartości >16.

Przykład:

```
mpls set dynamic-label-range=16-2000
```

3.7. LDP

mpls ldp print – wypisanie informacji o instancji LDP.

mpls ldp interface print – wypisanie interfejsów używanych do rozgłaszania komunikatów LDP,

mpls ldp neighbor print – wypisanie wykrytych sąsiadów protokołu LDP,

mpls ldp set enabled=yes lsr-id=<ID> transport-address=<ADRES_IP> – uruchomienie instancji protokołu LDP z podanymi parametrami *lsr-id* oraz *transport-address*. Parametr *lsr-id* ma format zbliżony do adresu IP i musi być unikalny dla każdego routera w systemie. Parametr *transport-address* określa, który z adresów IP posiadanych przez router będzie używany, jako źródłowy w wysyłanych przez niego komunikatach LDP. Często oba te parametry ustawia się na tę samą wartość (adres na interfejsie loopback – w naszym wypadku typu most).

Przykład:

```
mpls ldp set enabled=yes lsr-id=10.1.1.1 transport-address=10.1.1.1
```

mpls ldp interface add interface=<INTERFEJS> – dodanie interfejsu, na którym mają być odbierane i wysyłane znaczniki z użyciem protokołu LDP.

Przykład:

```
mpls ldp interface add interface=ether1
```

mpls ldp interface remove numbers=<NUMER(Y)> – usunięcie wpisów o podanych numerach z listy interfejsów używanych do rozgłaszania komunikatów LDP (patrz: *mpls ldp interface print*)

3.8. VPLS

interface vpls print – wypisanie wszystkich interfejsów VPLS

interface vpls add name=<NAZWA> remote-peer=<ADRES_IP> vpls-id=<ID>

mac-address = <ADRES_MAC> disabled=NO – dodanie interfejsu VPLS o zdefiniowanej nazwie (*NAZWA*) i adresie MAC (*ADRES_MAC*), który jest wejściem do jednokierunkowego tunelu VPLS prowadzącego do routera o adresie IP podanym w opcji *remote-peer*. *NAZWA* jest nazwą utworzonego interfejsu i ma znaczenia wyłącznie lokalne dla danego routera.

ID powinno być w postaci A:B, takie samo dla obu końców tunelu i unikalne dla każdego z tuneli pomiędzy tymi samymi 2 routerami. A i B mogą być dowolnymi, dodatnimi liczbami z przedziału od 0 do 4294967295. B można również podać w postaci przypominającej adres IP „b1.b2.b3.b4”, gdzie *bx* są liczbami 0..255.

Dzięki opcji *disabled=no* interfejs jest od razu włączony.

Przykład:

```
interface vpls add name=VPLS1 remote peer=10.2.4.2 vpls-id=10:0 mac-address=00:00:00:00:00:A1 disabled=no
```

interface vpls remove <NAZWA> – usunięcie interfejsu VPLS o nazwie. Zamiast nazwami można posłużyć się również numerami interfejsów.

Przykład:

```
interface vpls remove 1,3
interface vpls remove vpls1,vpls4
```

interface vpls monitor <NAZWA> once – wyświetlenie informacji o tunelu VPLS o podanej nazwie. Zamiast nazwą można również posłużyć się numerem interfejsu.

Przykład:

```
interface vpls monitor 1 once
interface vpls monitor vpls3 once
```

3.9. Traffic Engineering (TE)

mpls traffic-eng interface add interface=<NAZWA> bandwidth=<PRZEPIYWNOSC> – włączenie obsługi TE na interfejsie o podanej nazwie oraz określenie jego przepływności (w bps) na potrzeby ustalania trasy i dokonywania rezerwacji przez mechanizmy TE. Na potrzeby laboratorium zaleca się włączenie obsługi TE na wszystkich interfejsach routera.

UWAGA: polecenie nie powoduje rzeczywistej zmiany przepływności interfejsu, a jedynie przekazuje podaną informację mechanizmom TE.

Przykład:

```
mpls traffic-eng interface add interface=ether1 bandwidth=50000
```

Wskazanie interfejsu ether1 do procesu tworzenia tunelu TE oraz nakazanie potraktowania jego przepływności jako równej 50kb/s dla potrzeb wyznaczania trasy.

mpls traffic-eng tunnel-path add name=<NAZWA> use-cspf=yes – utworzenie definicji ścieżki, którą tunel TE ma przesyłać dane. Punktów początkowego i końcowego ścieżki nie definiujemy tutaj, lecz dopiero przy tworzeniu tunel TE (patrz polecenie poniżej).

Podana składnia tworzy ścieżkę w pełni polegającą na automatycznym wyznaczaniu trasy (CSPF). Z innymi parametrami, polecenie to pozwala na ręczne zdefiniowanie routerów, przez które ścieżka musi przebiegać.

Przykład:

```
mpls traffic-eng tunnel-path add name=dyn use-cspf=yes
```

Tworzy nową ścieżkę o nazwie dyn – ścieżka ta ma zdefiniowanych żadnych routerów przez które musi prowadzić, lecz jest w pełni dobierana automatycznie.

interface traffic-eng add name=<NAZWA> bandwidth=<PRZEPIYWNOSC> bandwidth-limit=100 primary-path=<ŚCIEŻKA> from-address=<ADRES_IP> to-address=<ADRES_IP> disabled = no record-route=yes – utworzenie tunelu TE o zadanej nazwie i wymaganej przepływności biegnącego po wcześniej zdefiniowanej ścieżce pomiędzy routerami o podanych adresach IP. Dodatkowa opcja *disabled=no* powoduje natychmiastowe włączenie tunelu, a *record-route=yes* pozwala na późniejszą weryfikację rzeczywistej trasy jego przebiegu. Parametr

bandwidth-limit=100 nakazuje wejściowemu routerowi ograniczać dopuszczalny ruch kierowany do tunelu do 100% zdefiniowanej przepływności.

Tunel taki będzie widoczny jako nowy interfejs routera (nazwa interfejsu będzie taka, jaką podaliśmy w powyższym poleceniu). Interfejs ten można wykorzystać jak każdy inny typ interfejsu – np. przypisując mu adres IP.

Dodatkowo, jeśli pomiędzy routerami o podanych w poleceniu adresach IP istnieje tunel VPLS, to jego ruch zostanie automatycznie skierowany na ten interfejs TE (a zatem przenoszony tunel VPLS ruch popłynie ścieżką ustaloną przez TE).

Przykład:

```
interface traffic-eng add name=te1 bandwidth=1000000 bandwidth-limit=100 primary-path=dyn from-address=1.1.1.1 to-address=2.2.2.2 disabled=no record-route=yes
```

Tworzy tunel TE pomiędzy routerami o adresach 1.1.1.1 i 2.2.2.2, o przepływności 1 Mbps i ruchu kierowanym wcześniej zdefiniowaną ścieżką o nazwie dyn.

interface traffic-eng monitor <NAZWA> once – wyświetlenie informacji o interfejsie (tunelu) TE o podanej nazwie. Zamiast nazwy można też podać numer interfejsu.

Przykład:

```
interface traffic-eng monitor te1 once
```

```
interface traffic-eng monitor 3 once
```

mpls traffic-eng path-state print – pozwala na wyświetlenie informacji na temat ścieżek TE przechodzących przez dany router,

mpls traffic-eng resv-state print – pozwala na wyświetlenie informacji na temat rezerwacji TE dokonanych na danym routerze,

mpls traffic-eng interface print – pozwala na wyświetlenie parametrów interfejsów danego routera, wykorzystywanych podczas wyznaczania ścieżek TE (np. całkowitej i pozostałej przepływności).

4. Dodatkowe objaśnienia

4.1. Interfejsy loopback

Interfejsy typu *loopback*, występujące na każdym z routerów w naszym systemie, pełnią rolę tzw. interfejsów *loopback*. Pełnią one szereg funkcji organizacyjnych i ułatwiają zarządzanie.

Protokół OSPF automatycznie pobiera z nich adres IP i ustawia wartość RouterID na równą temu adresowi, co pozwala uniknąć jej ręcznego konfigurowania i w związku z tym zmniejsza ryzyko błędów (gdy np. zapomnimy tego zrobić albo ustawimy RouterID tak samo na kilku routerach).

W przypadku protokołu LDP, to właśnie między adresami IP przypisanymi do interfejsów *loopback* ustanawiana jest sesja TCP dla celów wymiany znaczników MPLS. Stworzenie takiego interfejsu niepodłączonego do żadnej sieci (stąd maska /32) pozwala zminimalizować niestabilność działania protokołu LDP, która mogłaby wynikać np. z chwilowej niedostępności fizycznego interfejsu. Dodatkowo zapewnia to jasną identyfikację końców tunelu. Każdy router dysponuje wieloma interfejsami i adresami – bez konsekwentnej polityki, który z tych adresów podajemy jako końcówkę tunelu, trudno byłoby prześledzić naszą konfigurację.

4.2. Tablica routingu IP

Poleceniem **ip route print** jesteśmy w stanie wyświetlić kompletną tablicę routingu IP danego routera. Zawiera ona wszystkie trasy routingu IP, uzyskane przez komplet protokołów routingu aktywnych na danym routerze.

```
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
```

#	DST-ADDRESS	PREF-SRC	GATEWAY	DISTANCE
0	ADo 1.1.1.1/32		203.1.1.1	110
1	ADC 2.1.1.1/32	2.1.1.1	bridge1	0
2	ADo 3.1.1.1/32		203.1.3.1	110
3	ADo 203.1.0.0/24		203.1.1.1	110
4	ADC 203.1.1.0/24	203.1.1.2	ether2	0
5	ADo 203.1.2.0/24		203.1.3.1	110
			203.1.1.1	
6	ADC 203.1.3.0/24	203.1.3.2	ether1	0
7	ADC 203.3.0.0/24	203.3.0.1	ether3	0

Pole **DST-ADDRESS** określa docelową sieć IP, do której prowadzi dana trasa.

Flagi określają czy dana trasa jest:

- aktywna (A) – aby mogła zostać użyta, trasa musi być aktywna. Trasy nieaktywne pojawiają się, jeśli router zna kilka tras do tego samego DST-ADDRESS – aktywna i używana jest wtedy tylko najlepsza (wg różnych kryteriów, np. o najmniejszej wartości DISTANCE – patrz niżej).
- dynamiczna (D) – dopisana automatycznie przez router (np. prowadząca do sieci, do której należy dany router) lub protokół routingu dynamicznego,
- statyczna (S) – dopisana ręcznie przez administratora, odwrotność dynamicznej,
- bezpośrednio podłączona (C) – dany router posiada adres należący do tej sieci na którymś ze swoich interfejsów (należy do danej sieci),
- RIP (r) – uzyskana za pośrednictwem protokołu RIP,
- OSPF (o) - uzyskana za pośrednictwem protokołu OSPF,
- unreachable (U) – ruch zakwalifikowany do takiej trasy zostanie przez ruter odrzucony, a do nadawcy zostanie wysłany komunikat ICMP informujący o tym fakcie.

Jeśli mamy do czynienia z **trasą typu C**, w kolumnie **GATEWAY** podana jest nazwa interfejsu, na którym router posiada adres z danej sieci IP, a w kolumnie **PREF-SRC** podany jest sam ten adres.

W innym przypadku (**trasa nie C**) w kolumnie **GATEWAY** podany jest adres IP routera, pod który zostanie przesłany pasujący do danej trasy ruch (inaczej adres next-hop).

Kolumna **DISTANCE** obrazuje preferencję w wyborze danej trasy – jeśli kilka z nich prowadzi do tej samej sieci (DST-ADDRESS), aktywna staje się ta o najmniejszej wartości DISTANCE. Możliwe jest ręczne ustawianie tej wartości dla różnych tras, lecz najczęściej wartość ta ustawiana jest w zależności od pochodzenia trasy: dla tras C =0, dla tras S =1, dla tras o =110, dla tras r =120.

W przypadku protokołów routingu dynamicznego, można też wyświetlić ich wewnętrzną tablicę tras, na podstawie której tworzona jest potem zbiorcza, główna tablica routingu, opisana powyżej. Tablica taka zawiera najczęściej więcej informacji na temat tras, lecz obejmuje jedynie trasy uzyskane z użyciem danego protokołu routingu.

Polecenie ma najczęściej postać **routing <nazwa_protokołu> route print**.

Przykładowy wynik polecenia **routing ospf route print** przedstawiono poniżej.

```
[admin@MikroTik] > routing ospf route print
```

#	DST-ADDRESS	STATE	COST	GATEWAY	INTERFACE
0	1.1.1.1/32	intra-area	20	203.1.1.1	ether2
1	2.1.1.1/32	intra-area	10	0.0.0.0	bridge1
2	3.1.1.1/32	intra-area	20	203.1.3.1	ether1
3	203.1.0.0/24	ext-1	30	203.1.1.1	ether2
4	203.1.1.0/24	intra-area	10	0.0.0.0	ether2
5	203.1.2.0/24	intra-area	20	203.1.1.1	ether2
				203.1.3.1	ether1
6	203.1.3.0/24	intra-area	10	0.0.0.0	ether1
7	203.3.0.0/24	intra-area	10	0.0.0.0	ether3

Poza znanymi już kolumnami **DST-ADDRESS** i **GATEWAY**, widzimy również informacje o:

- **INTERFACE** – interfejsie, przez który osiągalny jest router wskazany jako następny przeskok w polu GATWAY.
- **COST** – wewnętrznym dla danego protokołu wskaźniku kosztu danej trasy. W przypadku istnienia kilku tras do tej samej sieci (DST-ADDRESS), dany protokół routingu zgłasza do głównej tablicy tylko trasy o najniższym koszcie.
- **STATE** – rodzaju trasy. W tym przykładzie widać trasy wewnątrz-strefowe OSPF (intra-area) oraz jedną trasę zewnętrzną (external type 1).

4.3. Sąsiedzi OSPF

Poniższa ilustracja przedstawia wynik polecenia *routing ospf neighbor print*. Widać, że dany router posiada dwóch sąsiadów OSPF, o podanych identyfikatorach (**router-id**), adresach IP (**address**) oraz stanie sąsiedztwa (**state**) typu „Full”.

```
[admin@MikroTik] > routing ospf neighbor print
0 instance=default router-id=10.10.1.1 address=10.10.1.1 interface=ether2
  priority=1 dr-address=10.10.1.2 backup-dr-address=10.10.1.1 state="Full"
  state-changes=4 ls-retransmits=0 ls-requests=0 db-summaries=0
  adjacency=4m46s

1 instance=default router-id=10.10.5.1 address=10.10.5.1 interface=ether1
  priority=1 dr-address=10.10.5.2 backup-dr-address=10.10.5.1 state="Full"
  state-changes=5 ls-retransmits=0 ls-requests=0 db-summaries=0
  adjacency=6m32s
```

4.4. OSPF Inter-area Route Summarization

Wewnątrz pojedynczej strefy (area) OSPF, każda z należących do niej sieci pojawia się jako pojedynczy wpis w tablicy tras protokołu, na każdym z routerów należących do danej strefy.

W przypadku podziału naszego systemu na więcej niż jedną strefę, możemy przeprowadzić tzw. sumowanie tras (route aggregation, route summarization). Powoduje ono, że informacje o wielu podsieciach obecnych w innej strefie OSPF, poza jej granicami rozsyłane są jako trasy zbiorcze (o krótszej masce), obejmujące kilka sieci składowych.

*Na przykład, jeśli wewnątrz strefy rozsyłane są trasy do istniejących w niej sieci **192.168.0.0/24** oraz **192.168.1.0/24**, to z wykorzystaniem sumowania sieci, na zewnątrz strefy informacja o nich pojawi się jako pojedyncza trasa do sieci **192.168.0.0/23**.*

Domyślnie sumowanie tras jest wyłączone. W celu jego włączenia, na routerze łączącym obie strefy (ABR – Area Border Router) należy ręcznie zdefiniować prefiksy, które mają być rozsyłane do strefy sąsiedniej.

*Na przykład, zakładając że sieci 192.168.0.0/24 i 192.168.1.0/24 zlokalizowane są w strefie o nazwie **area1**, polecenie pozwalające osiągnąć efekt opisany powyżej brzmi:*

routing ospf area range add area=area1 range=192.168.0.0/23

Zdefiniowanie takiego prefiksu spowoduje jego rozgłoszenie do strefy sąsiedniej, niezależnie czy w rozsyłającej strefie rzeczywiście znajdują się wszystkie składające się na niego podsieci. Jednocześnie, nie będzie tam rozgłaszana żadna szczegółowa trasa do sieci zawierającej się w tym prefiksie.

Na przykład, gdyby w powyższym przykładzie, zdefiniowano prefiks 192.168.0.0/22, to do strefy sąsiadującej z area1 nie mogłaby zostać rozgłoszona żadna z zawierających się w nim sieci, czyli np.

192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24 i 192.168.3.0/24. Jednocześnie rozgłoszony zostałby tam cały prefiks 192.168.0.0/22, pomimo że dwie ostatnie ze składających się na niego sieci (192.168.2.0/24 i 192.168.3.0/24) w rzeczywistości nie istnieją w strefie area1.

Fakt, iż router ABR może w ten sposób łatwo rozgłosić trasy do nieistniejących sieci, może przyciągnąć do danej strefy ruch, którego nie jest ona w stanie obsłużyć, gdyż w rzeczywistości nie wie gdzie go dostarczyć (sieć docelowa nie istnieje w strefie OSPF).

Aby pozbyć się takiego ruchu możliwie szybko, rozgłaszający dany prefiks router ABR tworzy u siebie wpis dotyczący całego rozgłaszanego prefiksu i nakazujący odrzucanie pasującego do niego ruchu (trasa typu U – patrz 4.2). Jeśli dany router ABR nie dysponuje inną, bardziej szczegółową trasą do sieci docelowej zawartej w tym prefiksie, to trasa typu U zostanie użyta, a ruch zostanie odrzucony. Jeśli natomiast taka trasa istnieje (została dopisana przez protokół OSPF, gdyż sieć rzeczywiście istnieje w strefie OSPF), to będzie ona miała większy priorytet (ze względu na dłuższą maskę) od wpisu nakazującego odrzucanie ruchu dla całego rozgłaszanego prefiksu i ruch zostanie dostarczony.

W powyższym przykładzie interesujący nas fragment tablicy routingu może wyglądać następująco:

	<i>DST-SRC</i>	<i>PREF-SRC</i>	<i>GATEWAY</i>	<i>DISTANCE</i>
<i>ADoU</i>	192.168.0.0/22			110
<i>ADo</i>	192.168.0.0/24	192.168.10.1	192.168.10.2	110
<i>ADo</i>	192.168.1.0/24	192.168.11.1	192.168.11.2	110

Wpisy z maskami /24 mają wyższy priorytet i zostaną użyte zamiast wpisu z maską /22, co pozwoli dostarczyć ruch do rzeczywiście istniejących sieci 192.168.0.0/24 i 192.168.1.0/24. W przypadku nieistniejących sieci 192.168.2.0/24 i 192.168.3.0/24 jedynym pasującym wpisem będzie wpis nakazujący odrzucenie ruchu.

Opisywany mechanizm pozwala na sumowanie wyłącznie tras do sieci należących do określonej strefy OSPF (intra-area). Istnieje również podobny mechanizm, External Route Summarization, dotyczący rozgłaszanych przez daną strefę tras zewnętrznych (external).

4.5. LDP

Poniższa ilustracja przedstawia wynik polecenia *mpls ldp neighbor print*.

Jak widać dany router posiada dwóch sąsiadów LDP. Kolumny **TRANSPORT** i **LOCAL-TRANSPORT** podają adresy IP (odpowiednio: sąsiada i naszego routera), między którymi protokół LDP nawiązał połączenie i wymienia dane.

Kolumna **PEER** pokazuje *lsp-id* routera, z którym LDP nawiązał połączenie.

Flaga D (Dynamic) oznacza, że router automatycznie wykrył danego sąsiada, a **O** (Operational), że sesja LDP z danym sąsiadem jest aktywna.

```

[admin@MikroTik] > mpls ldp neighbor print
Flags: X - disabled, D - dynamic, O - operational, T - sending-targeted-hello,
U - upls
#   TRANSPORT      LOCAL-TRANSPORT PEER      SEN
0  DO   20.20.20.5       20.20.20.2     20.20.20.5:0  no
1  DO   20.20.20.1       20.20.20.2     20.20.20.1:0  no

```

Protokół LDP przydziela, na danym routerze, znaczniki, których mają używać inne routery przy przesyłaniu mu pakietów IP przeznaczonych dla danego punktu docelowego (sieci IP). Następnie rozsyła tę informację, aby sąsiednie routery MPLS posiadały tę informację i mogły oznaczyć pakiet wysyłany do danego routera odpowiednim znacznikiem.

4.6. MPLS

Polecenie *mpls local-bindings print* przedstawia tabelę powyższych odwzorowań prefiks IP –

znacznik MPLS.

Widać wartości znaczników (kolumna **LABEL**) przydzielone do prefixów IP (**DST-ADDRESS**) i rozesyłane do sąsiednich routerów MPLS o podanych identyfikatorach lsr-id (**PEERS**).

Przykładowo wpis nr 4 oznacza, że dany router poinformował swojego sąsiada o lsr-id 20.20.20.2, że jeśli ten chciałby mu przesać pakiet adresowany do sieci IP 10.10.5.0/24 to ma go oznaczyć znacznikiem MPLS o wartości 20.

```
[admin@MikroTik] /mpls local-bindings> print
Flags: X - disabled, A - advertised, D - dynamic,
L - local-route, G - gateway-route, e - egress
# DST-ADDRESS LABEL PEERS
0 ADLe 20.20.20.1/32 inpl-null 20.20.20.2:0
1 ADLe 10.10.1.0/24 inpl-null 20.20.20.2:0
2 ADLe 10.10.4.0/24 inpl-null 20.20.20.2:0
3 ADG 20.20.20.5/32 19 20.20.20.2:0
4 ADG 10.10.5.0/24 20 20.20.20.2:0
5 ADG 20.20.20.2/32 21 20.20.20.2:0
6 ADG 10.10.2.0/24 22 20.20.20.2:0
```

Wpisy z flagą **egress (e)**, oznaczają, że router jest ostatnim routerem na trasie do danej sieci, który nie będzie już dokonywał przełączenia pakietu na podstawie znacznika MPLS, lecz po musi przekazać (pozbawiony znacznika) pakiet IP do bezpośrednio podłączonej do siebie sieci. Z powyższego względu otrzymany znacznik MPLS nie ma dla niego znaczenia, gdyż i tak zostanie usunięty, a końcowe dostarczanie odbywa się zawsze korzystając z klasycznych zasad routingu IP.

Aby oszczędzić poprzedniemu routerowi MPLS konieczności dołączania znacznika (który i tak nie zostanie użyty), a sobie konieczności jego zdejmowania, router wysyła do sąsiadów specjalną wartość znacznika (*implicit-null*), oznaczającą żądanie przesłania mu pakietu bez znacznika MPLS. Opisana powyżej optymalizacja nosi nazwę **penultimate hop popping**.

Polecenie ***mpls remote-bindings print*** przedstawia listę znaczników MPLS (kolumna **LABEL**) otrzymanych od sąsiadów (zidentyfikowanych po **lsr-id** – kolumna **PEER**), których dany router ma użyć, jeśli będzie przysyłał im pakiety IP, które mają docelowo być dostarczone do sieci IP o podanym adresie (**DESTINATION**). Podano też adres IP (**NEXTHOP**), pod który należy przesać dany pakiet po opatrzeniu do podanym znacznikiem.

```
[admin@MikroTik] /mpls remote-bindings> print
Flags: X - disabled, A - active, D - dynamic
# DST-ADDRESS NEXTHOP LABEL PEER
0 D 20.20.20.1/32 16 20.20.20.2:0
1 D 10.10.1.0/24 inpl-null 20.20.20.2:0
2 D 10.10.4.0/24 17 20.20.20.2:0
3 AD 20.20.20.5/32 10.10.1.2 18 20.20.20.2:0
4 AD 10.10.5.0/24 10.10.1.2 inpl-null 20.20.20.2:0
5 AD 20.20.20.2/32 10.10.1.2 inpl-null 20.20.20.2:0
6 AD 10.10.2.0/24 10.10.1.2 inpl-null 20.20.20.2:0
```

Jak już opisano powyżej, wartość znacznika (**LABEL**) równa **impl-null** oznacza żądanie przesłania pakietu IP bez znacznika MPLS, co sugeruje, że rozgłaszający ją router MPLS (o **lsr-id** podanym w kolumnie **PEER**) jest bezpośrednio podłączony do docelowej sieci IP.

Można też zauważyć, że tylko wpisy oznaczone flagą **active (A)** mają podaną wartość **NEXTHOP**, oznaczającą adres IP routera, któremu należy przekazać dany pakiet. Wpisy pozbawione wartości **NEXTHOP** są nieaktywne i przedstawiają alternatywne trasy do danej sieci docelowej (**DST-ADDRESS**), które nie są aktualnie wykorzystywane.

Polecenie ***mpls forwarding-table print*** pokazuje, w jaki sposób zamieniane są znaczniki przy przełączaniu pakietów IP przez dany router. Brak znacznika w kolumnie **OUT-LABEL** oznacza

wysłanie pakietu bez znacznika MPLS.

Patrząc na wpis nr 1 – Znacznik 19 zostaje zamieniony na 18, następnie pakiet ze znacznikiem 18 zostaje wysłany do punktu następnego routera na trasie (o adresie 10.10.1.2). Z wartości w kolumnie *DESTINATION* widać, że działanie to jest częścią procesu dostarczania pakietu IP pod adres 20.20.20.5.

```
[admin@MikroTik] /mpls forwarding-table> print
Flags: L - ldp, U - vpls, T - traffic-eng
#  IN-LABEL  OUT-LABELS  DESTINATION  I NEXTHOP
0  expl-null
1  L 19      18          20.20.20.5/32  e 10.10.1.2
2  L 20      18          10.10.5.0/24  e 10.10.1.2
3  L 21      18          20.20.20.2/32 e 10.10.1.2
4  L 22      18          10.10.2.0/24  e 10.10.1.2
```

Do śledzenia trasy pakietów w sieci MPLS używamy specjalnej wersji polecenia *traceroute*. Jest ona dostępna na routerach jako polecenie *tool traceroute <adres IP>*.

Wynik działania polecenia przedstawiono poniżej. W kolumnie **ADDRESS** widać adresy IP kolejnych interfejsów, przez które odbierany jest pakiet, a w kolumnie **STATUS** L=18 oznacza wartość znacznika, z jakim pakiet MPLS został odebrany przez podany interfejs. Puste pole **STATUS** oznacza odebranie pakietu bez znacznika.

```
[admin@MikroTik] > tool traceroute 20.20.20.5
# ADDRESS          RT1  RT2  RT3  STATUS
1 10.10.1.2         6ns  2ms  1ms  <MPLS:L=18,E=0>
2 20.20.20.5       2ns  1ms  1ms
```

4.7. Tunel VPLS pomiędzy dwoma routerami

Tunele VPLS pozwalają na przenoszenie ruchu typu Ethernet, a tym samym na uzyskanie komunikacji w warstwie 2 modelu ISO-OSI. Łączą one interfejsy VPLS – wirtualne interfejsy tworzone na routerach MPLS. Interfejsy te mogą następnie być używane tak samo, jak interfejsy typu Ethernet.

W celu utworzenia tunelu VPLS pomiędzy dwoma routerami MPLS, tworzymy na każdym z nich nowy interfejs VPLS, podając jednocześnie adres IP routera, na którym zlokalizowany będzie interfejs VPLS będący drugim końcem tunelu. Ponieważ pomiędzy dwoma routerami MPLS może istnieć więcej niż jeden tunel VPLS, podajemy dodatkowo identyfikator *vpls-id*, która musi być taka sama dla obu interfejsów będących końcówkami jednego tunelu – pozwoli to mechanizmom VPLS jednoznacznie powiązać te interfejsy ze sobą.

Ruch VPLS przenoszony jest przez sieć MPLS przy użyciu przełączania na podstawie znaczników, podobnie jak każdy inny ruch IP. Zawiera on jednak dodatkowo jeszcze jeden znacznik, nadawany przez router początkowy tunelu, przenoszony przezroczysto przez sieć MPLS i analizowany dopiero w końcowym routerze tunelu. Na podstawie tego znacznika, ruch kierowany jest do odpowiedniego interfejsu VPLS. Należy zaznaczyć, iż znacznik ten jest ustalany automatycznie i jego wartość nie musi być równa wartości identyfikatora *vpls-id* podawanemu przy tworzeniu interfejsu VPLS.

Wyświetlaniu stanu stworzonych interfejsów typu VPLS służy polecenie *interface vpls monitor <nazwa/numer> once*.

Local-label – znacznik VPLS (znacznik drugiego poziomu), z którym należy przesyłać pakiety do danego routera, aby ich zawartość trafiła do przeglądanego interfejsu VPLS.

Remote-label – znacznik VPLS (znacznik drugiego poziomu), z którym dany router powinien wysyłać pakiety IP pod adres routera obsługującego drugi koniec tunelu, aby ich zawartość trafiła do odpowiedniego interfejsu VPLS (tworzącego parę z obecnie przeglądany).

Obecność 2 powyższych wartości oznacza, iż tunel VPLS został pomyślnie utworzony.

Imposed-labels – wartość obu znaczników (MPLS i VPLS), które dodawane są do pakietu opuszczającego dany router w celu przeniesienia danych do drugiego końca tunelu. Pakiety te będą wysyłane przez dany router do jego sąsiada o adresie IP, który podany jest w linii **transport-nextthop**.

Linia **transport** podaje, która z tras routingu obecnych na danym routerze jest wykorzystywana, aby skierować tunelowany ruch w miejsce docelowe (tzn. w jakiej sieci znajduje się drugi koniec tunelu z punktu widzenia danego routera).

```
[admin@MikroTik] > interface vpls monitor 0 once
  remote-label: 25
  local-label: 26
  remote-status:
    transport: 20.20.20.1/32
  transport-nextthop: 10.10.5.2
  imposed-labels: 18,25
```

4.8. MPLS TE

Mechanizm tuneli MPLS TE pozwala uwzględnić w procesie wyznaczania ścieżki transmisji danych przez sieć MPLS dodatkowe wymagania, takie jak:

- odgórnie zdefiniowana część lub całość ścieżki,
- dodatkowe wymogi przy wyborze łączy (np. wymagana przepływność).

Tworzone tunele są jednokierunkowe.

Działanie powyższego mechanizmu opiera się na wykorzystaniu protokołu **RSVP** w celu weryfikacji proponowanej ścieżki oraz rezerwacji koniecznych zasobów.

Początkowy router ścieżki wysyła wiadomość RSVP Path do routera końcowego. Wiadomość ta jest następnie przekazywana przez kolejne routery MPLS, które zobowiązane są wziąć pod uwagę dodatkowe wymagania ścieżki przy wyborze kolejnego przeskoku. Gdy wiadomość RSVP Path dotrze do routera końcowego, odsyła on, tą samą trasą, którą przekazywana była wiadomość RSVP Path, wiadomość RSVP Resv do routera początkowego. Wiadomość ta rezerwuje, przechodząc przez kolejne routery na ścieżce, zasoby niezbędne do realizacji transmisji. Gdy router początkowy odbierze wiadomość RSVP Resv, ścieżkę można uznać za wyznaczoną.

Ze względu na możliwe zmiany w sieci (awarie, zmiany struktury, zmiany wykorzystania zasobów), powyższy proces jest periodycznie powtarzany w okresie istnienia ścieżki.

Należy zaznaczyć, iż dokonana powyżej rezerwacja zasobów na potrzeby wyznaczonej ścieżki, powoduje zmniejszenie się ich wartości pozostającej do dyspozycji kolejnych ścieżek. Np. jeśli przez łącze o przepływności 1 Mbps zostanie zarezerwowana ścieżka o wymaganej przepływności 600 kbps, to druga taka sama ścieżka nie będzie już mogła zostać zestawiona z użyciem tego łącza, gdyż jego pozostała przepływność to tylko 400 kbps.

W naszym przypadku, do wyznaczenia ścieżki wykorzystujemy algorytm **Constrained Shortest Path First (CSPF)**. Jego użycie pozwala na określenie wymaganej przepływności ścieżki, a także na odgórnie zdefiniowanie jej wymaganych fragmentów (wymuszenie wyznaczenia ścieżki przez określony zbiór routerów MPLS). W przypadku zastosowania CSPF, całą ścieżkę wyznacza początkowy router MPLS, po czym następuje z jej użyciem wymiana wiadomości RSVP Path/Resv w celu jej weryfikacji i dokonania rezerwacji zasobów.

Aby umożliwić uwzględnienie informacji o przepływnościach łączy przy wyznaczaniu ścieżek TE, należy najpierw zdefiniować przepływności wykorzystywanych łączy. Zadanie to należy wykonać ręcznie, poleceniem **mpls traffic-eng interface add ...** dla wszystkich łączy w systemie. Należy zaznaczyć, iż użycie powyższego polecenia nie ogranicza przepływności łączy między routerami, lecz tylko przepływność braną pod uwagę przy wyznaczaniu ścieżki przez mechanizm TE.

Kolejnym krokiem jest zdefiniowanie ścieżki TE poleceniem **mpls traffic-eng tunnel-path add ...**

Pozwala ono na określenie czy do jej wyznaczenia ma być stosowany algorytm CSPF oraz ewentualne wskazanie wymaganych routerów MPLS, przez które ścieżka musi prowadzić. Konkretna ścieżka (lista routerów i łączy, przez które przebiegać będzie transmisja) nie jest jeszcze w tym momencie wyznaczana.

Ostatnim krokiem jest samo utworzenie tunelu TE poleceniem *interface traffic-eng add ...* wskazując, jakiej (wcześniej zdefiniowanej) ścieżki TE ma używać tunel oraz podając dodatkowe wymagania (np. przepływność). Powoduje to wyznaczenie i rezerwacje konkretnej ścieżki przez sieć MPLS oraz utworzenie interfejsu typu TE na jej początkowym routerze. Interfejs ten można wykorzystać jak każdy inny typ interfejsu – np. przypisując mu adres IP.

Dodatkowo, jeśli pomiędzy początkowym i końcowym routerem tunelu TE istnieje tunel VPLS, to jego ruch zostanie automatycznie skierowany przez tunel TE (a zatem przenoszony tunel VPLS ruch popłynie ścieżką ustaloną przez TE).

Należy pamiętać, że tunele TE są JEDNOKIERUNKOWE. W celu realizacji transmisji dwukierunkowej ścieżkami wyznaczonymi przez TE, należy zestawić 2 tunele TE.

Poniżej przedstawiono przykładowy wynik polecenia *interface traffic-eng monitor 0 once*.

Widać, że tunel został zestawiony (**primary-path-state: established**) z użyciem wcześniej zdefiniowanej ścieżki o nazwie stat (**active-path: stat**).

Przy tworzeniu nie nakazano wyznaczenia ścieżki zapasowej (**secondary-path-state: not-necessary**).

Reserved-bandwidth podaje jaką przepływność zarezerwowaliśmy dla ścieżki, a **rate-limit** określa, do jakiej przepływności router wejściowy ma ograniczać ruch nią kierowany.

Pola **explicit-route** i **recorded-route** pozwalają zaobserwować trasę, którą biegnie stworzony tunel TE.

Explicit-route podaje wyznaczoną dla tunelu ścieżkę, wymieniając po kolei wszystkie interfejsy routerów (wejściowe i wyjściowe) przez które ma przechodzić ruch.

Recorded-route podaje wynik rzeczywistego badania trasy, którą ruch został skierowany – podaje interfejsy wejściowe routerów na trasie oraz wartości znacznika MPLS z którym pakiety tunelu są odbierane na danym interfejsie (w nawisach kwadratowych).

```
[admin@R1] /interface traffic-eng> monitor tel
      tunnel-id: 3
      primary-path-state: established
      primary-path: stat
      secondary-path-state: not-necessary
      active-path: stat
      active-lspid: 1
      active-label: 20
      reserved-bandwidth: 100.0kbps
      rate-limit: 120.0kbps
      rate-measured-last: 0bps
      rate-measured-highest: 0bps
```

Poza opisanym powyżej, monitorowaniu działania mechanizmów MPLS TE mogą służyć następujące polecenia:

mpls traffic-eng path-state print – pozwala na wyświetlenie informacji na temat ścieżek TE przechodzących przez dany router,

mpls traffic-eng resv-state print – pozwala na wyświetlenie informacji na temat rezerwacji TE dokonanych na danym routerze,

mpls traffic-eng interface print – pozwala na wyświetlenie parametrów interfejsów danego routera, wykorzystywanych podczas wyznaczenia ścieżek TE (np. całkowitej i pozostałej przepływności).