



**TITANIA**<sup>®</sup>  
NETWORKS LTD

# Ingeniería de tráfico

Introducción a la ingeniería de tráfico con MPLS  
cuando no es suficiente el “best path”

mum

MEXICO ON APRIL 16 - 17, 2018

2018 Alfredo Giordano, Titaniania Networks Limited



# Alfredo Giordano

- Trainer Certificado MikroTik.
- Consulente certificado para MikroTik, Cisco, Ub\*\*\* especializado en el desarrollo de ISP y WISP.
- Proporcionando soluciones con MikroTik desde el 2006.
- Trabajando en telecomunicaciones desde el 2001.
- Administrador de varios LIR
- Graduado en Ingenieria Electronica en el Politecnico de Torino, ITA y universidad de Illinois en Chicago, USA.

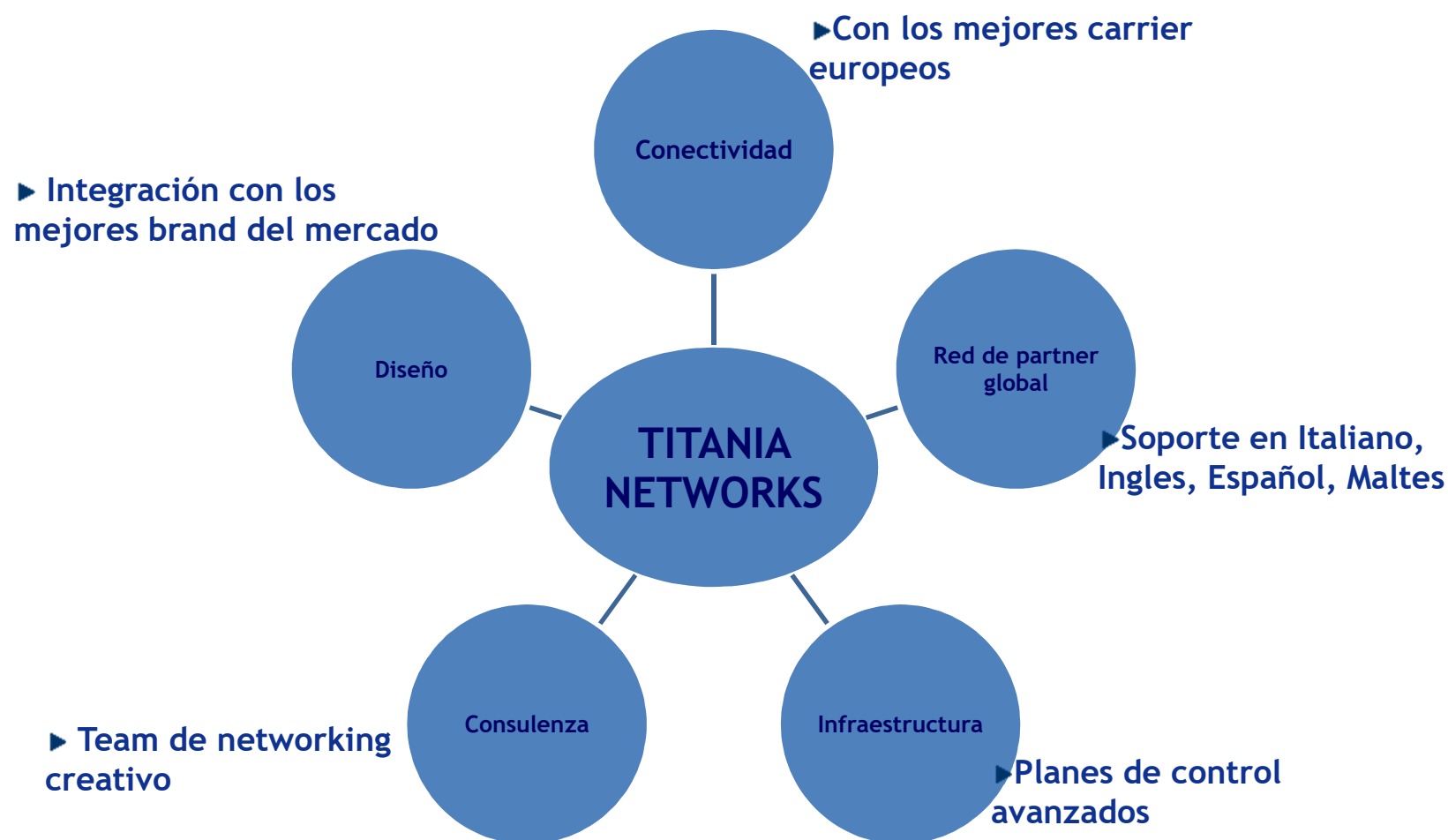


# Titania Networks

- Fundada en el 2015 en Dublín, Irlanda
- Responde a necesidades de negocios con criticidad en redes.
- Principales servicios:
  - Consultoría de red misión critica
  - Diseño de ISP
  - Network Training
- Áreas de operación:
  - Europa
  - América Latina
  - India



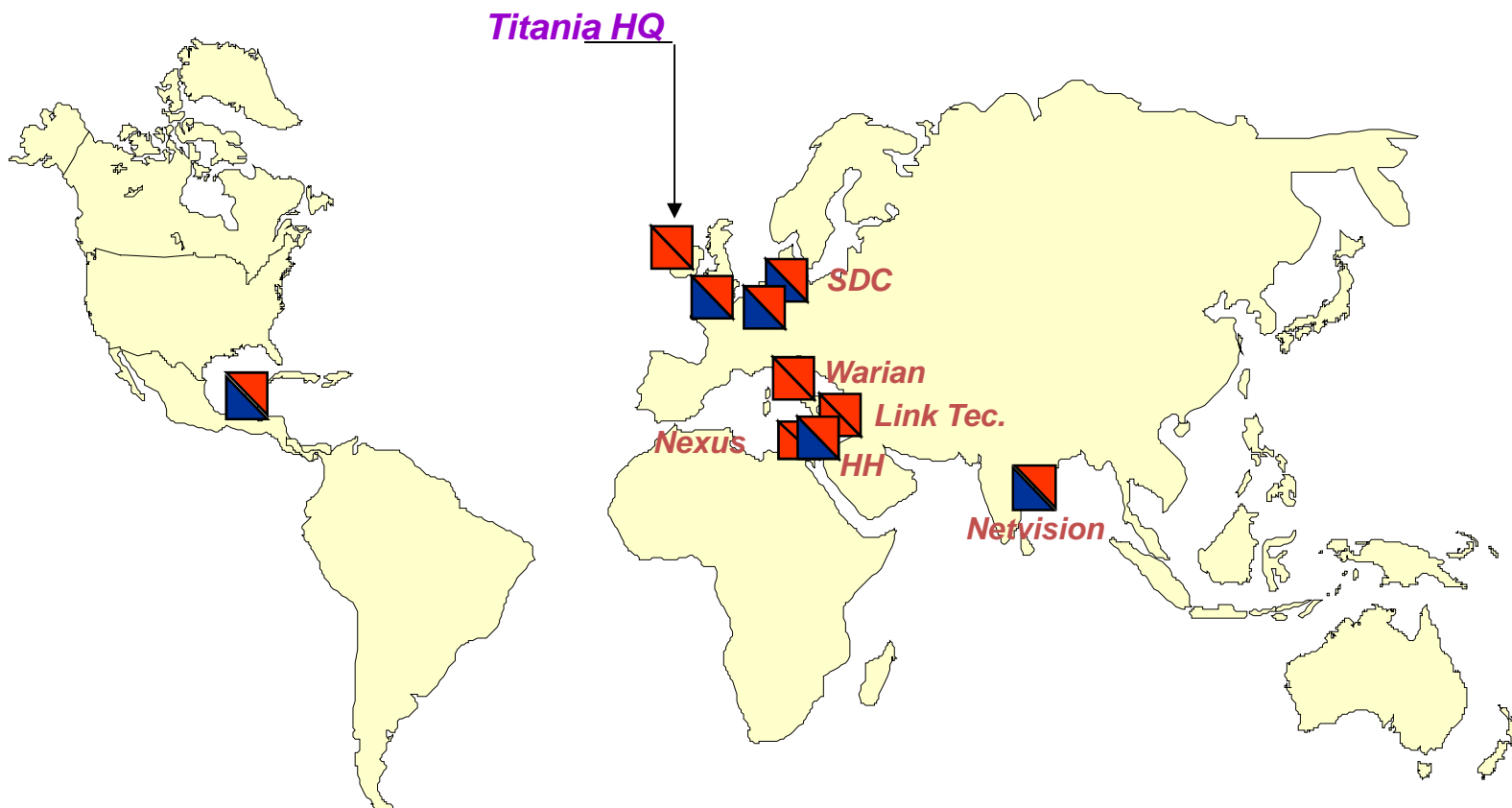
# Operaciones



- Con un solo punto de contacto.



# Clientes



-  Consulting Customer
-  Partner with Infrastructure facility



# Objetivos

- Introducir principios básicos de la ingeniería de tráfico.
- Como empezar a usar MPLS y RSVP para ingeniería de tráfico.
- Un ejemplo de como sacarle provecho en aplicaciones del mundo real con las habilidades de red obtenidas.



# A quien se dirige

- ISP's y Operadores que implementan (o piensan implementar) políticas de ingeniería de tráfico con MikroTik RouterOS

## Disclaimer

- Esta presentación se propone como una introducción. Se omiten deliberadamente algunos elementos con el fin de simplificar.



# Agenda

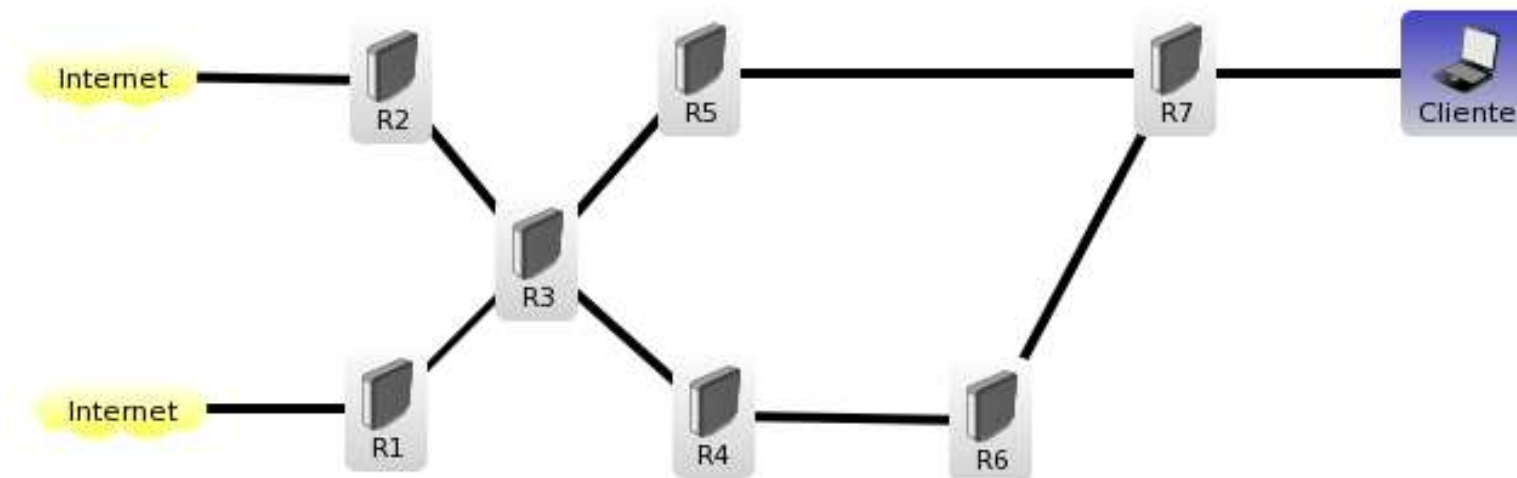
- Introducción a la ingeniería de tráfico
  - Mas allá de TE
  - Ingeniería de tráfico sin TE
- Componentes de MPLS-TE:
  - Routing Engine
  - MPLS, OSPF, LDP, RSVP
- Túneles MPLS-TE
  - Interfaces
  - Como enviar tráfico por medio de los túneles
- Un ejemplo práctico
  - conclusiones
- Créditos





# Introducción a la ing. de trafico

- Ya tengo mi anillo y ahora?

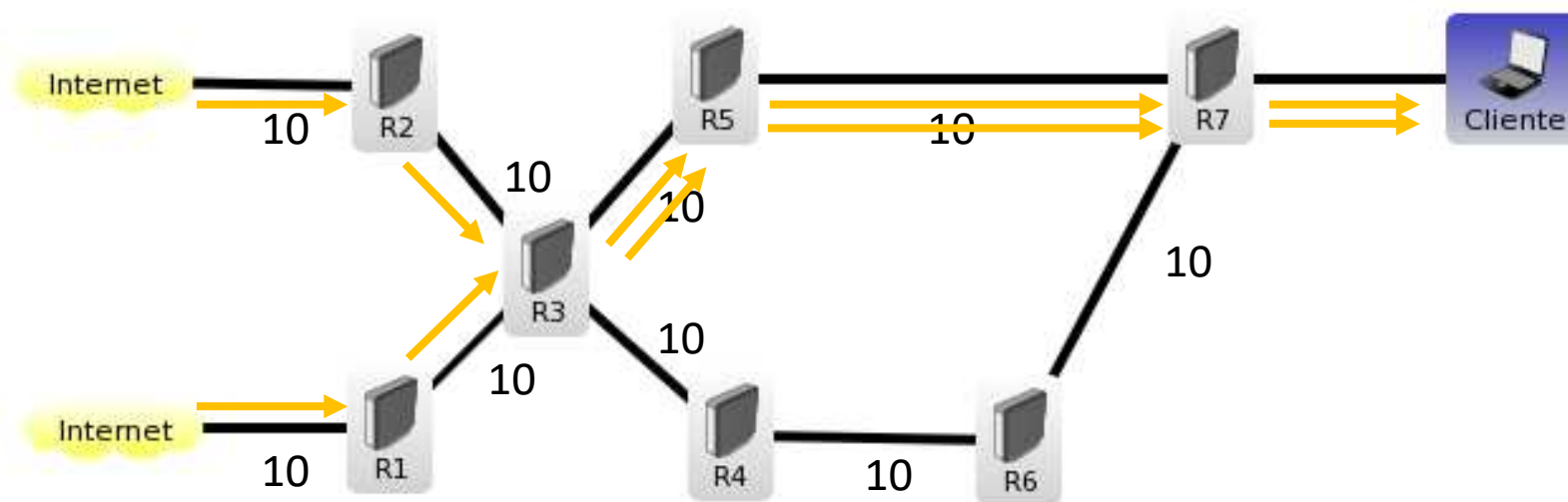


- Un problema mayor de las redes es la tarea de ajustar el trafico para hacer un mejor uso del ancho de banda.



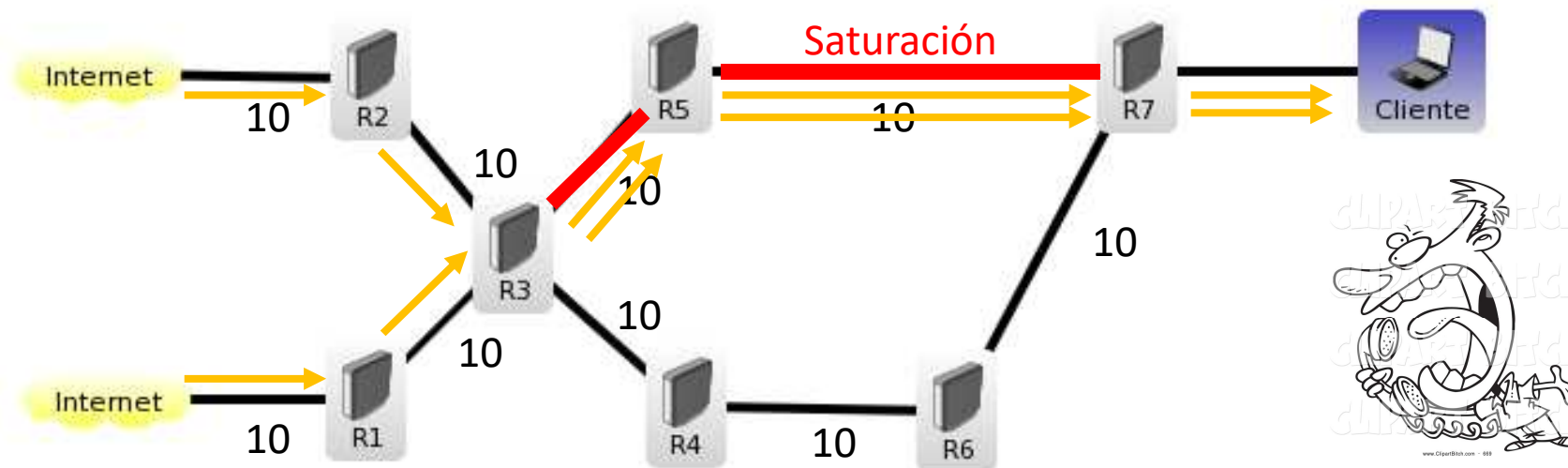
# Introducción a la ing. de trafico

- En las redes IP los paquetes suelen seguir el camino mas corto. Esto suele ocasionar que algunos enlaces se saturen mientras otros estén poco utilizados.





# Introducción a la ing. de trafico



- La congestión influye en retardos, jitter, perdidas y por lo tanto en la calidad de servicio.



# Introducción a la ing. de trafico

- Las causas de la congestión son:
  - Insuficiencia de recursos.
  - Uso ineficiente de los recursos debido al protocolo de enrutamiento.
- Las posibles soluciones incluyen
  - Añadir enlaces de mas capacidad.
  - Añadir mas capacidad a los enlaces por medio de la ingeniería de trafico.



# Introducción a la ing. de trafico

- Las técnicas de Ingeniería de trafico intentan adaptar flujos de red a recursos físicos de la misma.
- Es decir se intenta que no haya recursos sobre utilizados (mejor conocidos como cuello de botella) y recursos subutilizados.



# Mas alla de T.E.

Independientemente de la técnica que se utilice el fin de la ingeniería de tráfico es desarrollar las estructuras para proporcionar calidad en el servicio de red a los usuarios.

Por lo tanto distinguimos en:

- Ingeniería orientada a tráfico
  - minimizar retardos, perdidas, etc..
- Ingeniería orientada a recursos
  - Optimización de ancho de banda.



# Mas alla de T.E.

En general la ingeniería de trafico nos otorga la capacidad de:

- Maximizar el ancho de banda agregado.
- Mejorar el control en caso de fallas.
- Re direccionar caminos alrededor de los cuellos de botella.



# Ingeniería de tráfico sin T.E.

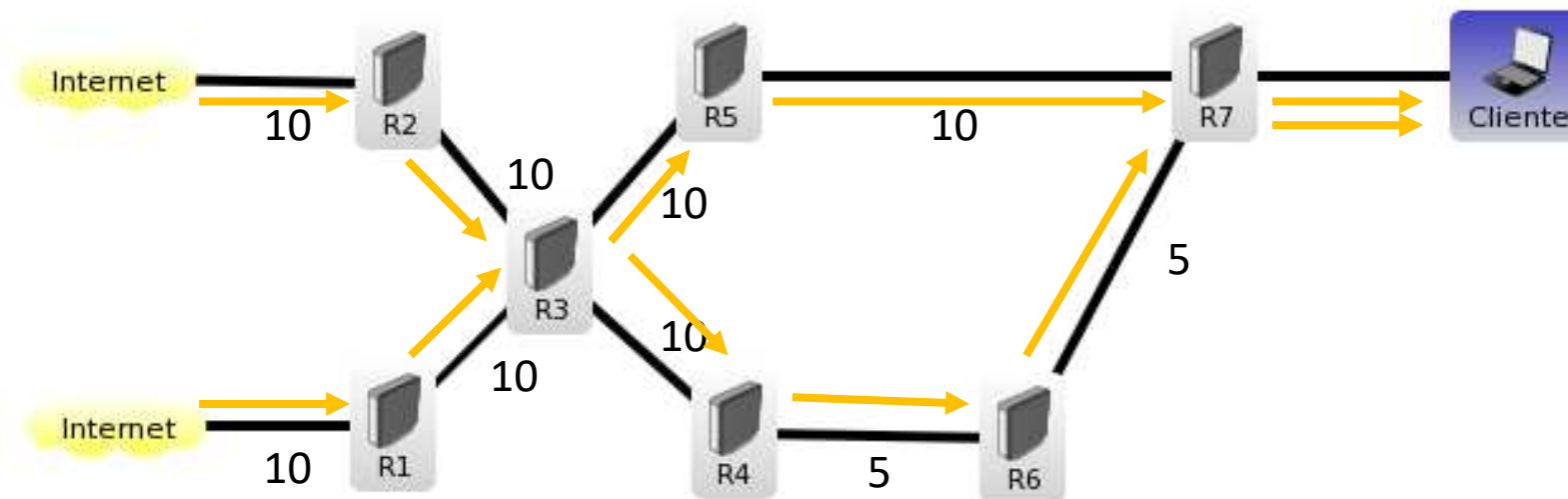
- Se puede con una correcta planeación del IGP
  - Ajustes a la métrica y costos
  - Por lo general implica replanear toda la red
  - En algunos casos se pueden formar anillos si se trabaja con métrica asimétrica.





# Ingeniería de tráfico sin T.E.

- Ejemplo de ingeniería de tráfico con IGP



\*suponiendo una métrica simétrica



# Componentes de MPLS-TE

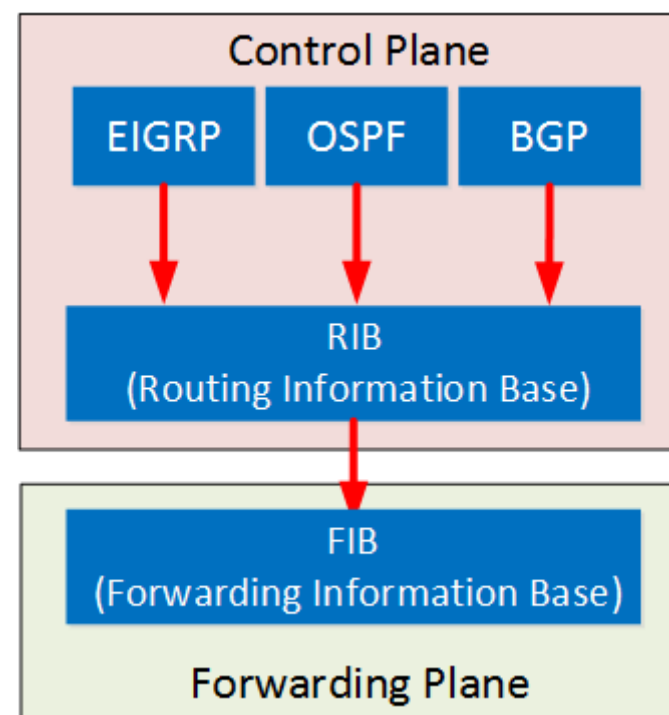
- Routing Engine
  - Determina como elegir la mejor ruta hace a un destino.
- Distribución de la información (control plane)
  - OSPF: distribuye la informacion de rutas usada para el calculo de los LSP.
- Packet Forwarding
  - MPLS: responsable de encaminar el trafico.
- Selección de camino y señalización
  - LDP y RSPV.



# Routing Engine

- El kernel encamina los paquetes basandose en la FIB (Forward Information Base)

Las FIB se construyen basándose en la RIB (Routing Information Base)





# Routing Engine

- La RIB contiene la información de del next-hop por cada prefijo:

Routes	Nexthops	Rules	VRF
DAS	▶ 0.0.0.0/0		10.50.0.1 reachable WAN
DAC	▶ 10.1.1.0/24		LAN-PUB reachable
DAC	▶ 10.50.0.0/20		WAN reachable
DAC	▶ 192.168.15.0/...		LAN-PRIV reachable
DAC	▶ 192.168.253.0...		vlan20 reachable
DAC	▶ 192.168.254.0...		vlan10 reachable



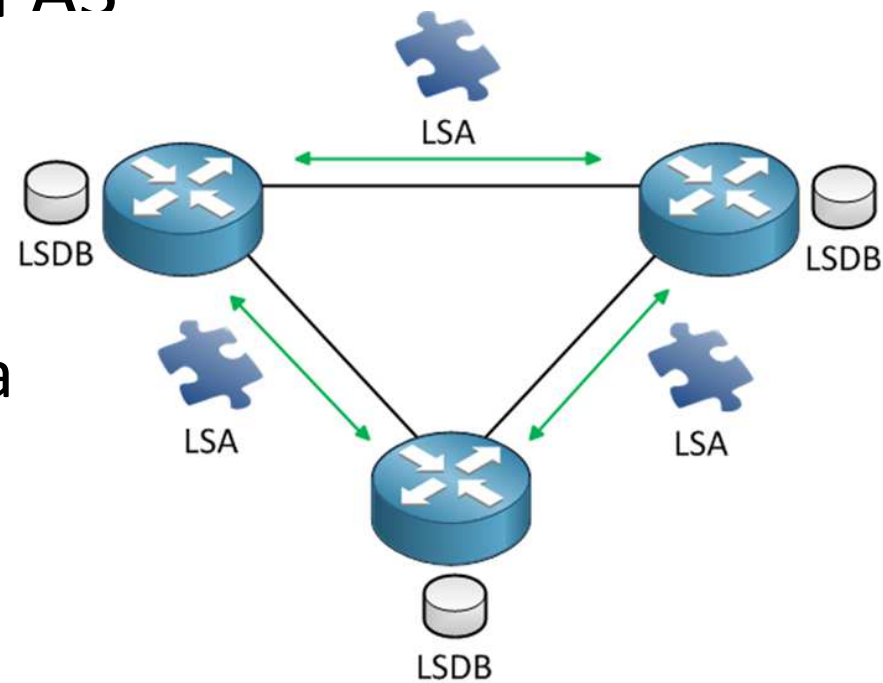
# OSPF

- Open Shortest Path First
  - Es un protocolo de routing dinámico que se clasifica en los IGP (Interior Gateway protocol)
  - Es un protocolo link state organizado por áreas en el cual cada router intercambia la información de sus redes por medio de mensajes específicos (LSA)
  - Tiene soporte a los path de túneles TE por medio de la área opaca.



# OSPF

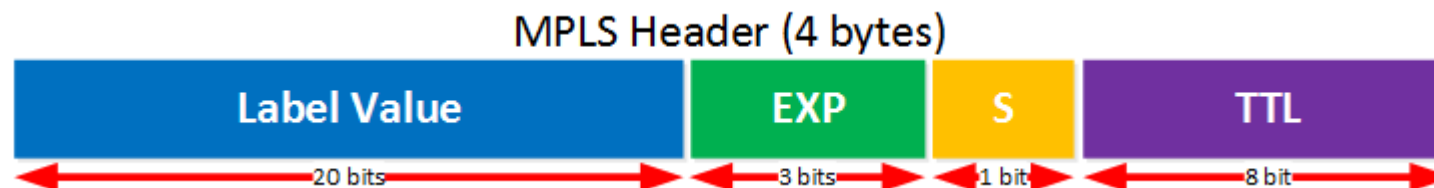
- En practica OSPF para determinar la tabla de routing se basa sobre el estado de los enlaces de cada router. Dicho estado es comunicado a los demás routers del AS
- Así como en un rompecabezas todas los LSA constituyen el LSDB
- Todo Router ospf maneja un identificador univoco RouterID





# MPLS

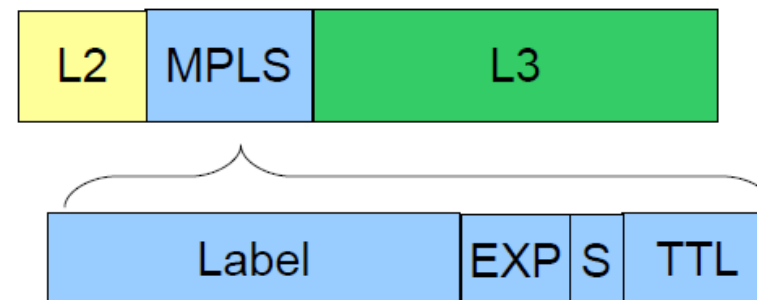
- Multi Protocol Label Switching (RFC3032)
  - Es un mecanismo estandar de transporte de datos que opera entre capa 2 y capa 3 del modelo OSI
    - Requiere OSI Capa 2 y OSI Capa 3 Funcionando
  - Asigna a cada destino una etiqueta que permite la comutacion entre los routers del camino sin volver a mirar al destino del paquete.
  - Aplicaciones: QoS, Ingenieria de trafico, VPNs





# MPLS

- MPLS funciona añadiendo un encabezado entre la capa 2 y la capa 3 que contiene una o mas etiquetas (label stack).
- Estas etiquetas se establecen cuando el paquete entra a la nube MPLS y se remueven cuando sale de ella.

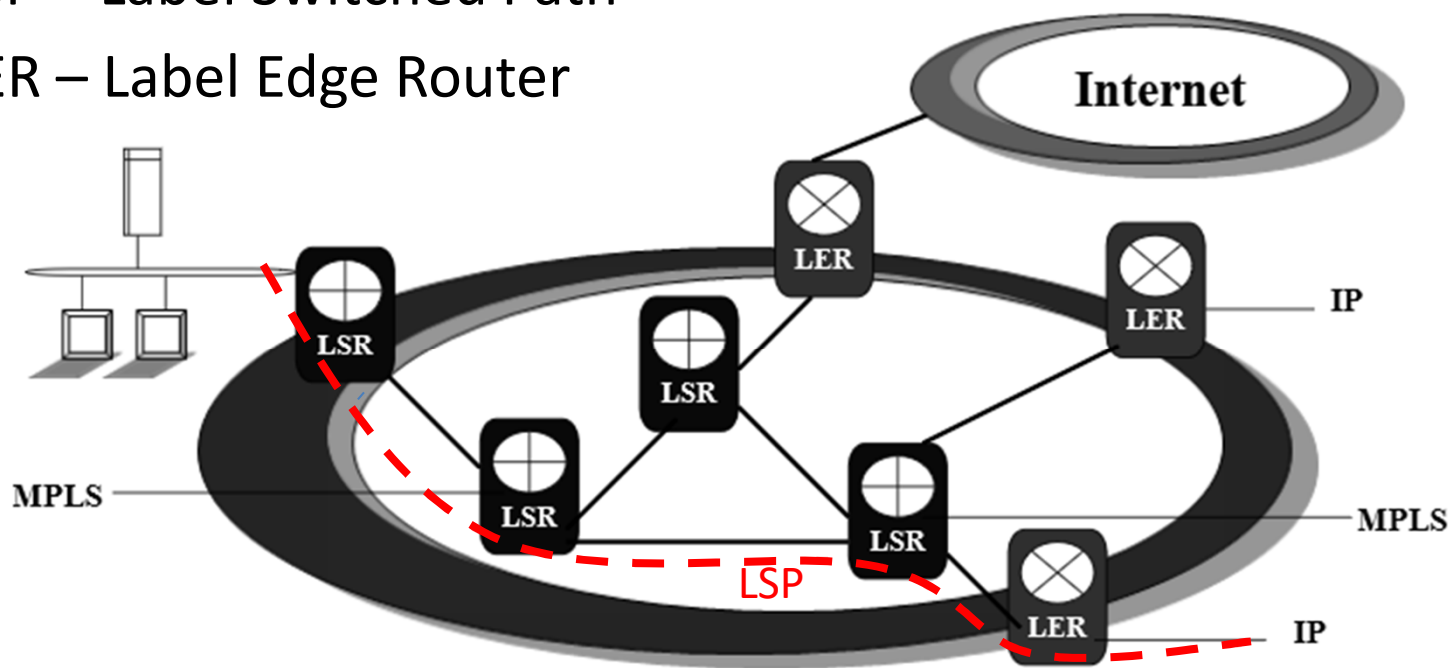






# MPLS

- LSR – Label Switching Router
- LSP – Label Switched Path
- LER – Label Edge Router





# MPLS

- Las etiquetas permiten establecer un LSP (un VC para los que le gusta ATM) y la comutacion sucede unicamente basada en la etiqueta.
- Al recibir un paquete MPLS el LSR:
  - Examina la etiqueta y la interfaz.
  - Consulta la tabla de etiquetas.
  - Determina la nueva etiqueta y la interfaz de salida.



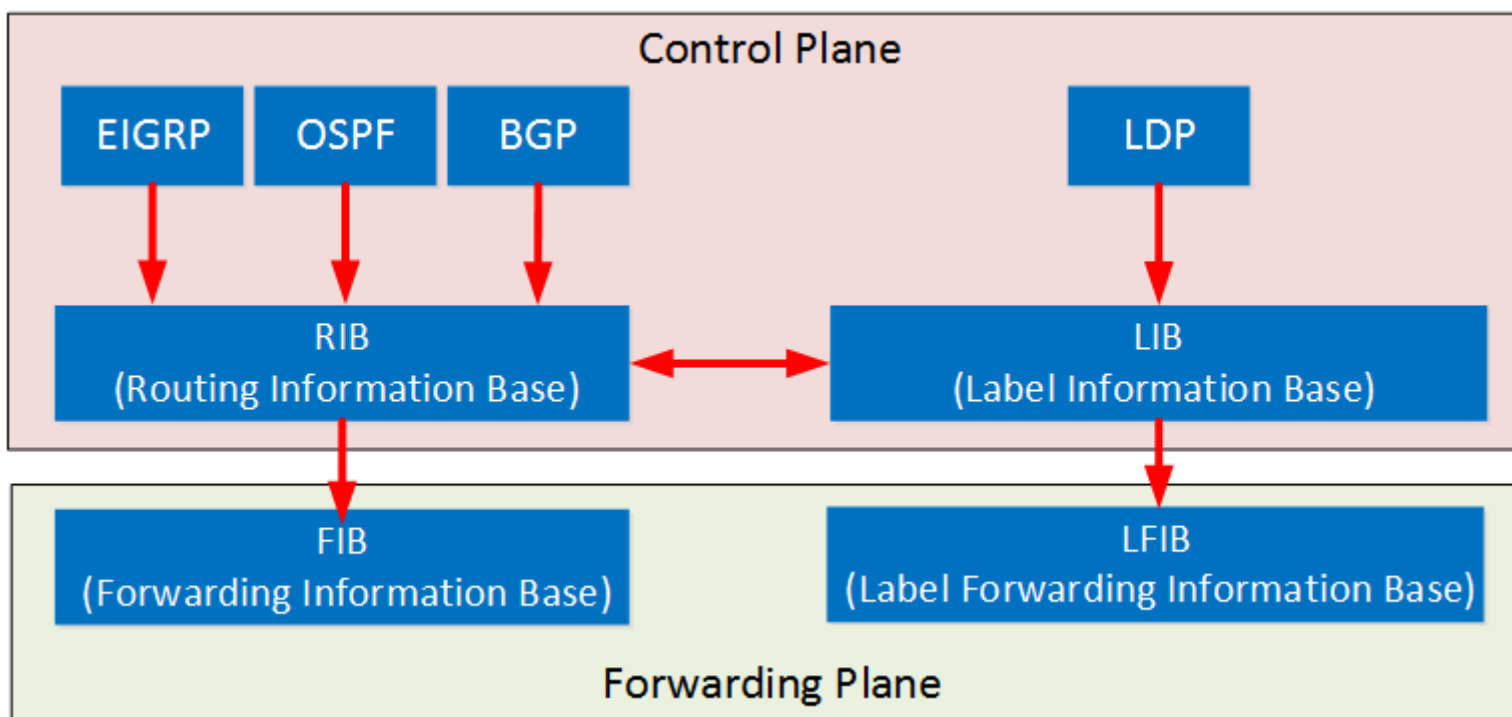
# LDP

- Label Distribution Protocol
  - MPLS puede utilizar LDP para mantener una base de datos de etiquetas entre routers.
  - En general utilizado para señalar la configuración de una nube MPLS.



# LDP

- LDP influye en el control plane (usa LIB y LFIB)





# LDP

- La LIB contiene la información de las etiqueta para todas las rutas activas:

	Dst. Address	Label	Nexthop	Peer	Path
DA	0.0.0.0/0	impl-null	172.31.0.101	176.58.82.131:0	empty
DA	10.50.1.0/29	impl-null	172.31.0.101	176.58.82.131:0	empty
DA	10.50.3.0/29	1291	172.31.0.101	176.58.82.131:0	empty
DA	10.50.4.0/29	1547	172.31.0.101	176.58.82.131:0	empty
DA	10.50.5.0/29	2391	172.31.0.101	176.58.82.131:0	empty
DA	10.50.6.0/29	1638	172.31.0.101	176.58.82.131:0	empty
DA	10.50.7.0/29	1416	172.31.0.101	176.58.82.131:0	empty



# RSVP

- Resource REserVation Protocol
  - Es un protocolo de transporte de MPLS.
  - RSVP-TE es una extensión de RSVP.
  - RSVP-TE mantiene una base de datos que asocia las etiquetas MPLS a los caminos.
  - La base de datos RSVP-TE actualiza los caminos por medio de la extensión del IGP en nuestro caso la área opaca de OSPF.



# Mas información

- Para seguir investigando:
- MPLS - <https://tools.ietf.org/html/rfc3031>
- OSPF - <https://tools.ietf.org/html/rfc2328>
- RSVP-TE - <https://tools.ietf.org/html/rfc3209>
- LDP - <https://tools.ietf.org/html/rfc5036>



# Túneles MPLS-TE

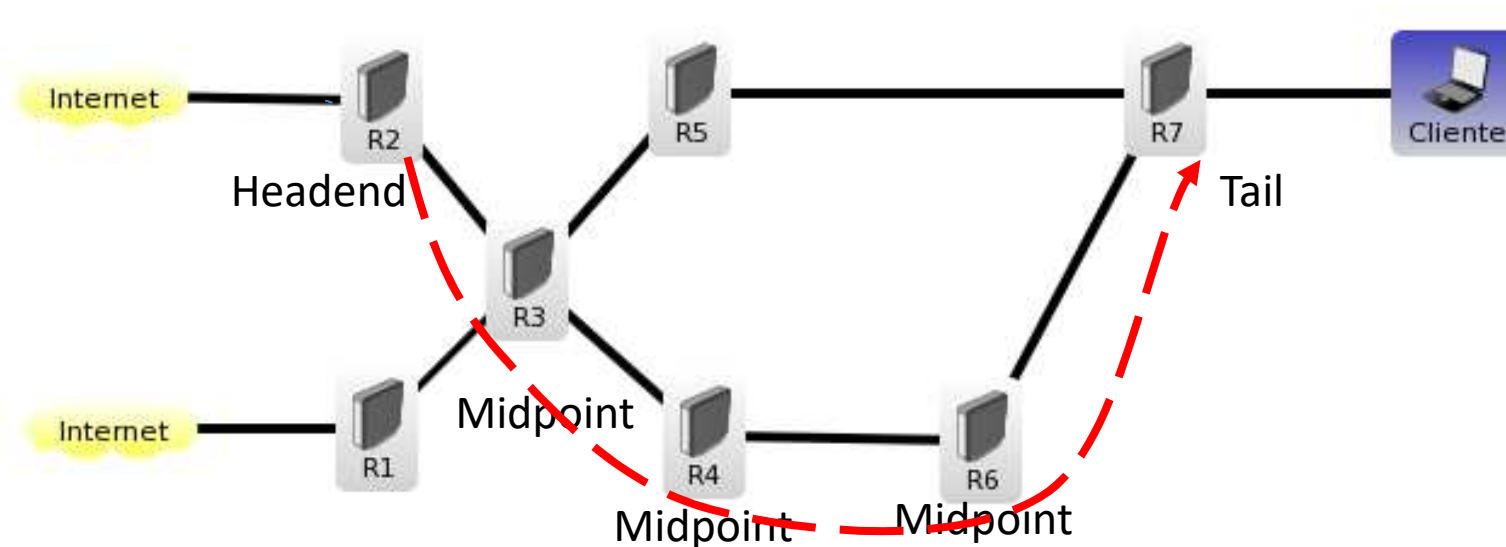
- Los túneles de TE implementan una interfaz de salida asociada con un camino definido internamente a la red. Se usan para poder enviar el tráfico a un camino predeterminado en la nube MPLS.
  - El camino de red es configurado y no necesariamente el que calcula el IGP
- Los túneles TE son una interfaz lógica unidireccional que representa un link de un router a otro
  - Una vez configurado el túnel se asigna una etiqueta que corresponde a dicho túnel





# Túneles MPLS-TE

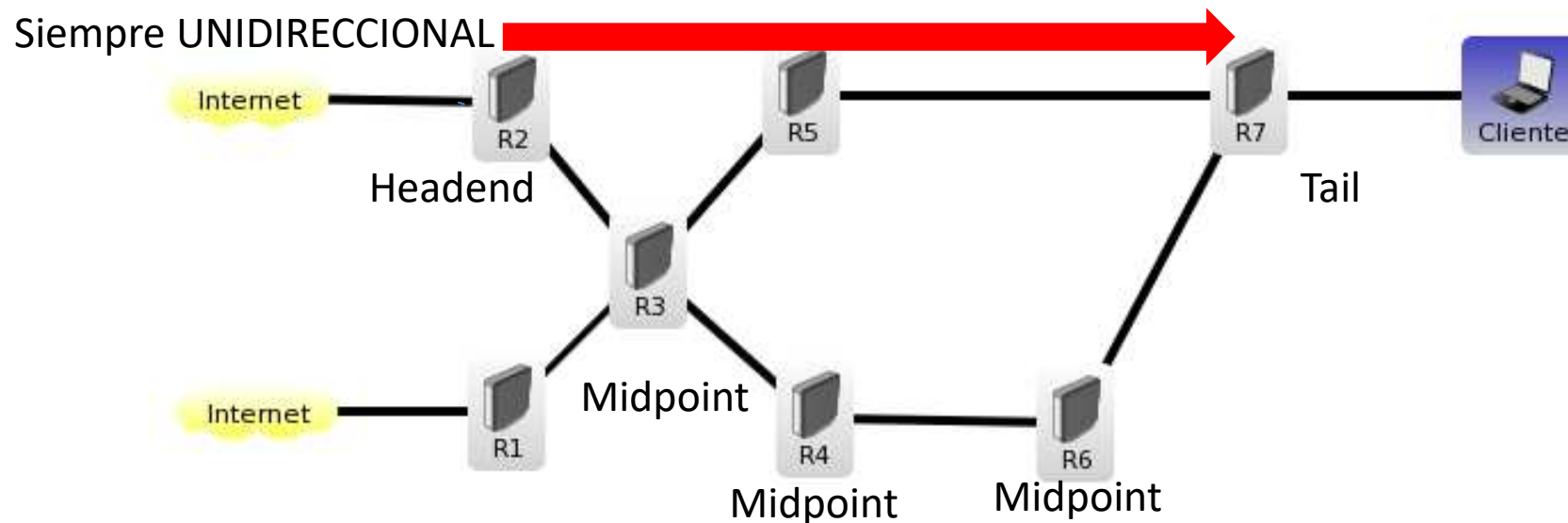
- Nomenclatura
  - Headend: el router donde se configura el túnel
  - Tail: Router de destino del túnel
  - Midpoint: Todos los routers en el camino





# Túneles MPLS-TE

- CSFP – Constrain Shortest Path First
  - el router Headend elabora el camino mas corto basado en una serie de atributos fijos como:
    - Disponibilidad de recursos a lo largo del camino
    - Parámetros del usuario (prioridad etc..)





# Túneles MPLS-TE

- Se tiene que configurar una ruta para enviar tráfico a la interfaz de túnel
  - A veces eso ocasiona confusión sobre cual ruta es de tráfico y cual de control
- El tráfico de regreso puede ser enrutado en una interfaz en sentido contrario
  - O simplemente se puede dejar al IGP



# Túneles MPLS-TE

- Sin embargo hay una importante característica que nos facilita la vida:
  - VPLS utiliza automáticamente el camino de TE si Headend y Tail son los mismos

  [https://wiki.mikrotik.com/wiki/Manual:TE\\_Tunnels](https://wiki.mikrotik.com/wiki/Manual:TE_Tunnels)

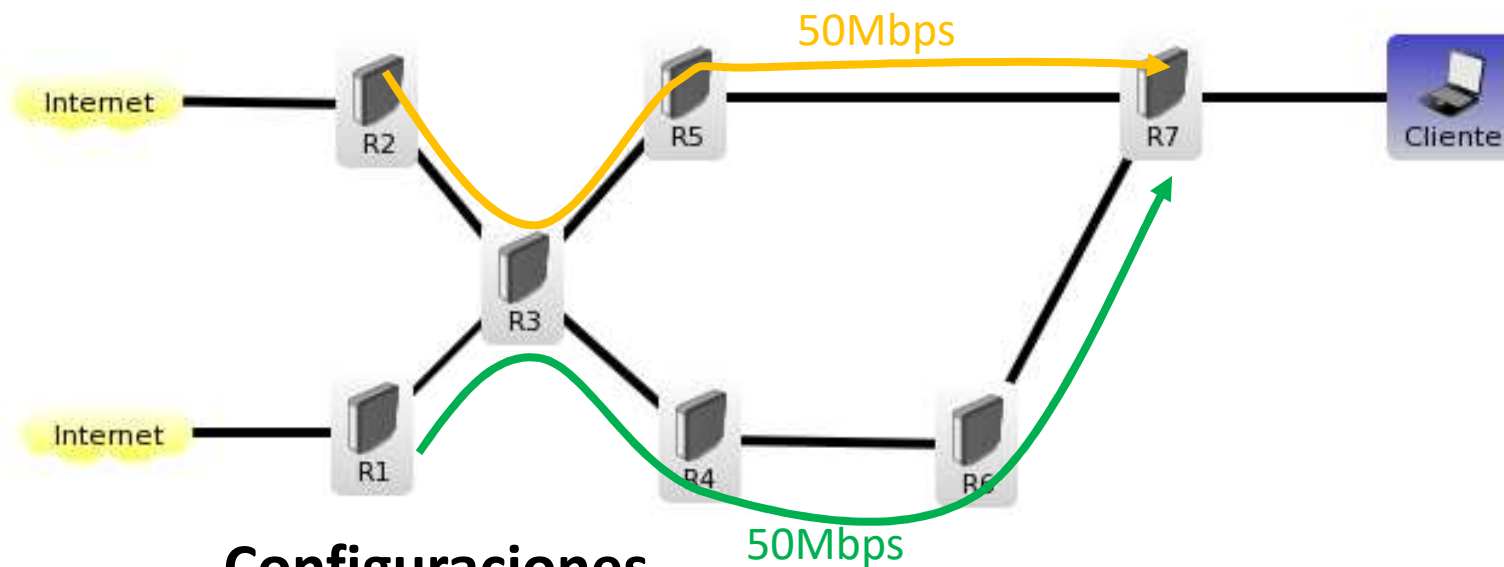
## Forwarding traffic onto TE tunnels

Additionally, several types of traffic can be forwarded onto TE tunnel automatically, |

- traffic for VPLS interfaces, if remote endpoint of VPLS pseudowire is the same as TE tunnel endpoint.



# Un ejemplo practico



## Configuraciones

- Generales
- Loopbacks
- IP
- OSPF
- MPLS
- LDP
- VPLS
- RSVP-TE
- TE interface



# Un ejemplo practico

- **Generales**

```
/system reset-configuration no-defaults=yes  
/system identity set name=RX
```

- **Loopbacks**

```
/interface bridge add fast-forward=no name=loopback protocol-mode=none
```

- **IP**

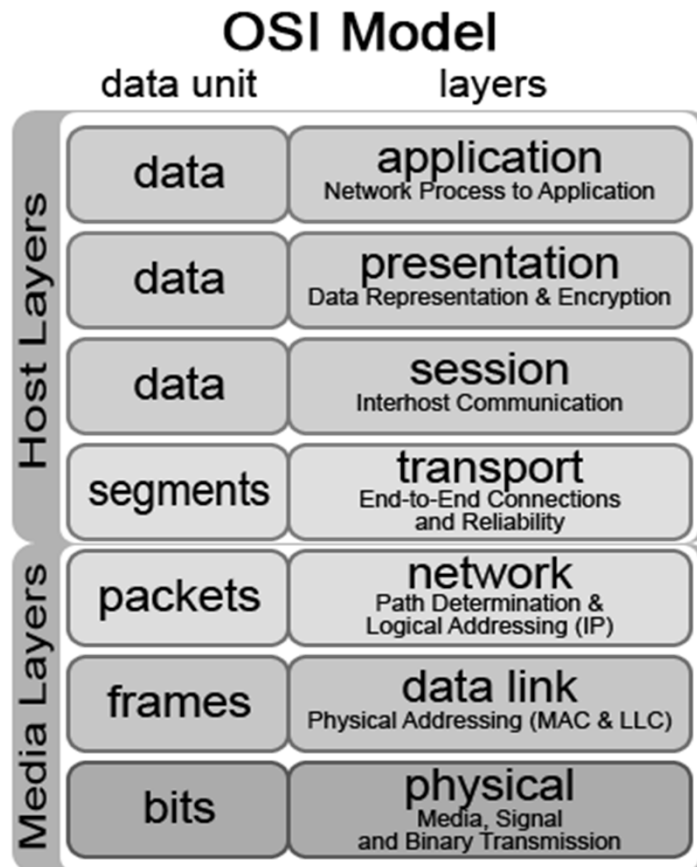
```
/ip address add address=10.0.0.X/32 interface=loopback comment=Loopback  
/ip address add address=172.20.XY.X/24 interface=ehter1 comment=RX_RY
```

**(Ej. R3)**

```
/ip address add address=10.20.0.3/32 interface=loopback comment=Loopback  
/ip address add address=172.20.13.3/24 interface=ehter1 comment=R3_R1  
/ip address add address=172.20.23.3/24 interface=ehter2 comment=R3_R2  
/ip address add address=172.20.35.3/24 interface=ehter3 comment=R3_R5  
/ip address add address=172.20.34.3/24 interface=ehter4 comment=R3_R4
```



# PAUSA - checkpoint



- A poco pensaron que se nos pudiera olvidar?



# Checkpoint

- L1 check

Interface List

Interface Interface List Ethernet EoIP Tunnel IP Tunnel GRE Tunnel VLAN VRRP Bonding LTE

+ - ✓ ✗ 📁 🔍

	Name	Type	Actual MTU	L2 MTU	Tx	Rx
	::: ->R2					
R	ether5	Ethernet	1500	1598	369.6 kbps	
R	Loopback	Bridge	1500	1598	14.9 Mbps	

14 items (1 selected)

- L2 check

Neighbor List

Neighbors Discovery Interfaces

🔍

Interface	IP Address	MAC Address	Identity	Platform
ether5	172.31.0.107	E4:8D:8C:0F:9D:03	R5	MikroTik





# Checkpoint

- L3 check

Ping (Running)

General | Advanced

Ping To: 172.20.34.1

Interface:

ARP Ping

Packet Count:

Timeout: 1000 ms

Seq #	Host	Time	Reply Size	TTL	Status
10	172.20.34.1	0ms	50	64	
11	172.20.34.1	0ms	50	64	
12	172.20.34.1	0ms	50	64	
13	172.20.34.1	0ms	50	64	
14	172.20.34.1	0ms	50	64	

15 items | 15 of 15 packets received | 0% packet loss | Min: 0 ms | Avg: 0 ms



# Un ejemplo practico

- OSPF

```
/routing ospf instance
```

```
set [ find default=yes ] mpls-te-area=backbone mpls-te-router-id=loopback router-id=10.20.0.X
```

```
/routing ospf interface add interface=all network-type=point-to-point passive=no
```

```
/routing ospf network add area=backbone network=172.20.0.0/16
```

```
/routing ospf network add area=backbone network=10.20.0.0/24
```

- MPLS

```
/mpls set dynamic-label-range=1000-0
```

```
/mpls interface set 0 mpls-mtu=1596
```

- LDP

```
/mpls ldp set enabled=yes lsr-id=10.20.0.X transport-address=10.20.0.X
```

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

```
/mpls ldp interface add interface=ether2
```



# Un ejemplo practico

- VPLS

- R1-Headend

```
/interface vpls add disabled=no l2mtu=1508 name=R1-R7_50M remote-peer=10.20.0.7 vpls-id=1:7
```

- R2-Headend

```
/interface vpls add disabled=no l2mtu=1508 name=R2-R7_50M remote-peer=10.20.0.7 vpls-id=2:7
```

- R7-Tail

```
/interface vpls add disabled=no l2mtu=1508 name=R7-R1_50M remote-peer=10.20.0.1 vpls-id=1:7
```

```
/interface vpls add disabled=no l2mtu=1508 name=R7-R2_50M remote-peer=10.20.0.2 vpls-id=2:7
```

- RSVP

- R1, R2

```
/mpls traffic-eng interface add interface=ether1 bandwidth=100Mbps
```

- R3

```
/mpls traffic-eng interface add interface=ether1 bandwidth=100Mbps
```

```
/mpls traffic-eng interface add interface=ether2 bandwidth=100Mbps
```

```
/mpls traffic-eng interface add interface=ether3 bandwidth=100Mbps
```

```
/mpls traffic-eng interface add interface=ether4 bandwidth=100Mbps
```

- R4,R5,R6,R7

```
/mpls traffic-eng interface add interface=ether1 bandwidth=100Mbps
```

```
/mpls traffic-eng interface add interface=ether2 bandwidth=100Mbps
```



# Un ejemplo practico

- RSVP

- R1-Headend

```
/mpls traffic-eng tunnel-path add name=IGP use-cspf=yes  
/mpls traffic-eng tunnel-path add name=VIA_R6 use-cspf=no \  
hops=10.20.0.3:strict,10.0.20.4:strict,10.0.20.6:strict,10.0.20.7:strict
```

- R2-Headend

```
/mpls traffic-eng tunnel-path add name=IGP use-cspf=yes  
/mpls traffic-eng tunnel-path add name=VIA_R5 use-cspf=no \  
hops=10.20.0.3:strict,10.0.20.5:strict,10.0.20.7:strict
```

- TE Interface

- R1

```
/interface traffic-eng add bandwidth=50Mbps name=TE_R1-R7 to-  
address=10.20.0.7 primary-path=VIA_R6 \  
secondary-paths=IGP record-route=yes from-address=10.20.0.1
```

- R2

```
/interface traffic-eng add bandwidth=50Mbps name=TE_R2-R7 to-  
address=10.20.0.7 primary-path=VIA_R5 \  
secondary-paths=IGP record-route=yes from-address=10.20.0.2
```



# Un ejemplo practico

- Lo opcional que falta para dejar el ejemplo completamente funcionando

- R1,R2

```
/ip dhcp-client add interface=ether5 add-default-route=yes  
comment=internet  
/ip address add address=10.X.1.254/24 interface=RX-R7_50M  
/ip pool add name=dhcp_pool ranges=10.X.1.100-10.X.1.200  
/ip dhcp-server add address-pool=dhcp_pool authoritative=after-2sec-  
delay disabled=no interface=RX-R7_50M lease-time=6h name=dhcp  
/ip dhcp-server network add address=10.X.1.0/24 dns-  
server=8.8.8.8,8.8.4.4 gateway=10.X.1.254
```

- R7

```
/interface bridge add name=transport_R1  
/interface bridge add name=transport_R2  
/interface bridge port add interface=R7-R1_50M bridge=transport_R1  
/interface bridge port add interface=R7-R2_50M bridge=transport_R2  
/interface bridge port add interface=ether5 bridge=transport_R1  
/interface bridge port add interface=ether4 bridge=transport_R1
```



# Conclusiones

Con la tecnología presentada se pueden armar complejas configuraciones en una red IP tomando el control de los caminos de red y optimizando los recursos sin perder la redundancia.

Sugerencias:

- Usar un esquema de direcciones IP simple y efectivo.
- Configurar los túneles VPLS con los mismos endpoint de las interfaces TE
- Configurar los tunnel path estáticos con: use-cspf=no hops=<explicit hop config>
- Usar los tunnel path dinámicos con: use-cspf=yes y ningún salto especificado como backup



# Créditos

- La presentación anterior MPLS-VPLS para ISP MUM MX14
- Los varios RFCs
- Imágenes “routing engine” ©networklessons
- [https://wiki.mikrotik.com/wiki/Manual:TE\\_Tunnels](https://wiki.mikrotik.com/wiki/Manual:TE_Tunnels)
- <http://www.timoid.org/wp-content/uploads/2015/05/MikroTik-Presentation-Tim-Warnock-For-Release.zip>
- [https://eva.fing.edu.uy/pluginfile.php/73921/mod\\_folder/content/0/9%20-%20MPLS/MPLS1.pdf](https://eva.fing.edu.uy/pluginfile.php/73921/mod_folder/content/0/9%20-%20MPLS/MPLS1.pdf)

Por su atención

¡GRACIAS!

Bienvenidas preguntas y sugerencias

*ag@tnxe.net*

