



Segment Routing

- Δημήτρης Καλογεράς, Phd,
- ΕΠΙΣΕΥ/ICCS

Αναφορές:

Nokia / Apricot 2017 @ khatri

Cisco Fosdem 2018 @ camarillo

Agenda



1. Εισαγωγή - Αναδρομή
2. Χρήσεις (Use cases) και εφαρμογή
3. SRv6
4. Srv6 Network Programming
5. Μερικές Εφαρμογές ακόμα
6. Emulations
7. Linux



Εισαγωγή



MPLS: Αναδρομή (1)

- Δύο πρωτόκολλα για label distribution με MPLS
- LDP ή/και RSVP-TE
 - LDP για απλότητα – με επεκτάσεις για fast re-route (loop-free alternates [LFAs])
 - RSVP-TE για TE και FRR
- Για επεκτασιμότητα του MPLS δημιουργήθηκαν:
 - LDPoRSVP
 - Seamless MPLS: Labeled-BGP με LDP
- Traffic engineering: RSVP-TE με δυνατότητες++:
 - BGP/IGP shortcuts, PW (T-LDP/BGP), VPLS (LDP/BGP), VPRN (BGP), MVPN (BGP/mLDP/P2MP RSVP)
- Προβλήματα:
 - TE λύσεις δεν κλιμακώνουν όταν απαιτείται FULL MESH LSP (x PE) granularity/dynamicity
 - unidirectional
 - R-LFA αρκετά περίπλοκο (dynamic T-LDP signaling)

MPLS: Αναδρομή (2)



	LDP	RSVP-TE
Overview	Multipoint to point	Point to point
Operation	Simple	LSP per destination/TE- path
Dependencies	Relies on IGP	Relies on IGP TE
LBL allocation	Local significant per node (interface)	Local significant per node (interface)
Traffic Engineering	No	Yes
Scaling	1 LBL per node (interface)	$N \times (N-1)$
Fast Reroute	LFA, LFA Policies, RLFA - <100% coverage	Link/Node protection (detour/facility) – 100% coverage
Multicast	mLDP	P2MP RSVP
IPv6	Extensions required	Extensions required

Ποιο είναι το πρόβλημα ?

SCALE

- Αύξηση δυναμικής κίνησης με απαίτηση για traffic engineering (TE)
π.χ. DC East-West
- RSVP-TE η μοναδική ευρέως διαδεδομένη λύση για TE
- Δεν υπάρχει LDP-TE
- LDP Fast ReRoute (FRR) μπορεί να χρησιμοποιηθεί σε ορισμένες τοπολογίες αλλά εξαρτάται από την εκάστοτε υφιστάμενη τοπολογία.

RSVP-TE

- Πλεονεκτήματα:
 - Source Routed protocol - iLER έχει πλήρη έλεγχο για την δημιουργία LSP μέχρι το προορισμό
 - Presence of strong FRR and TE capabilities
- Μειονεκτήματα:
 - Soft-state ; Χρειάζεται μηχανισμό refresh : refresh reduction (RFC2961) ; Με μηνύματα aggregate αλλά όχι # soft-states
 - καταναλώνει CPU cycles και μνήμη όταν υπάρχει μεγάλος αριθμός tunnels



Στόχος του Segment Routing

- Πρόβλημα προς λύση: ΤΕ σε κλίμακα και σε διακριτότητα
- Προτεινόμενη Λύση: segment routing

Η δυνατότητα ενός κόμβου να καθορίζει μια διαδρομή (unicast forwarding path) διαφορετική από την τυπική (normal shortest path) διαδρομή (source routing)

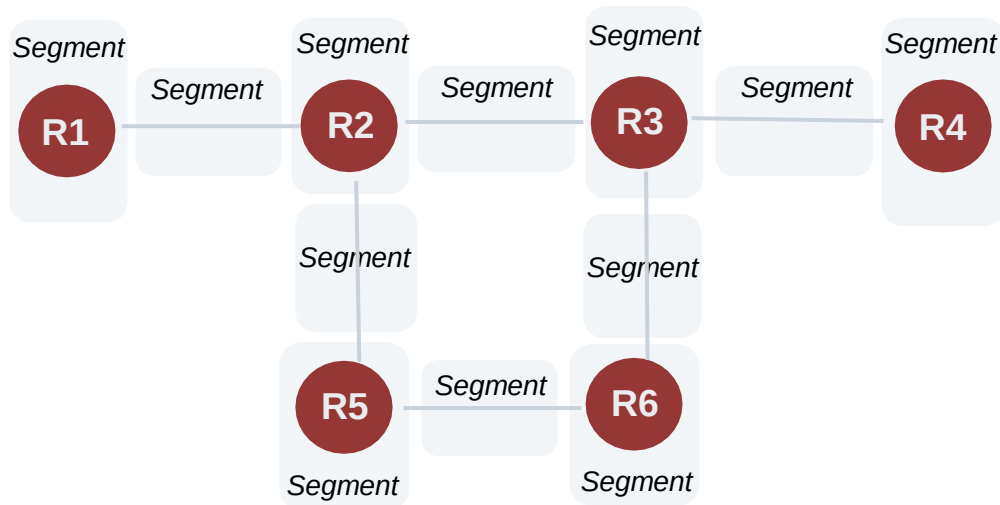


IETF SPRING working group

- SPRING (Source Packet Routing in Networking) Working Group μελετά:
 - IGP-based MPLS tunnels ΧΩΡΙΣ την προσθήκη σηματοδοσίας signaling protocol
 - Η δυνατότητα για tunnel services (VPN, VPLS, VPWS) από το ingress PE στο egress PE με/χωρίς explicit path, και χωρίς **ΠΡΟΣΘΕΤΟ** forwarding plane/control plane σε ενδιάμεσους κόμβους.
 - Fast Reroute
 - οποιαδήποτε τοπολογία, υπολογισμός πριν την εγκαθίδρυση της backup διαδρομής χωρίς πρόσθετη σηματοδοσία.
 - Υποστήριξη shared-risk constraints, υποστήριξη προστασίας link/node, υποστήριξη αποφυγής βρόγχου (micro-loop avoidance)
 - Κανένα νέο dataplane !! Υποστήριξη MPLS ...ή με ελάχιστη τροποποίηση νέο IPv6 hdr

Εισαγωγή Segment Routing

- Segment Routing προσφέρει ένα μηχανισμό με tunnels για source routing.
- Paths (διαδρομές) κωδικοποιούνται σαν *ακολουθία* (sequence) από τοπολογικά υπο-μονοπάτια τα οποία ονομάζονται segments, τα οποία ανακοινώνονται από τα πρωτόκολλα δρομολόγησης (link-state) IS-IS ή/και OSPF).



Κωδικοποίηση Segment Routing tunnels

- Ένα Segment Routing (SR) tunnel, περιέχει είτε ένα μοναδικό segment ή μια λίστα από segment κωδικοποιημένο ως:
 - Ένα μοναδικό MPLS label ή μια διατεταγμένη λίστα από hops αναπαριστώμενη από stack MPLS labels (ΚΑΜΙΑ αλλαγή σε σχέση με MPLS data-plane !!!)→ και άρα στο HW
 - Μια μοναδική διεύθυνση IPv6, μια διατεταγμένη λίστα από hops, αναπαριστώμενη από δ/νσεις IPv6 ΕΝΤΟΣ ενός NEOY IPv6 Extension header (Segment Routing Header-SRH)!!
- Η λίστα segment list μπορεί να αναπαριστά μια νέα τοπολογική διαδρομή ή μια υπηρεσία (service).

Τα segments μπορεί να θεωρηθούν σε ένα σύνολο εντολών από τον ingress PE : “*πήγαινε στο κόμβο D χρησιμοποιώντας shortest path*” ή “*πήγαινε στο κόμβο D χρησιμοποιώντας το/τα link/node/explicit-route L*”



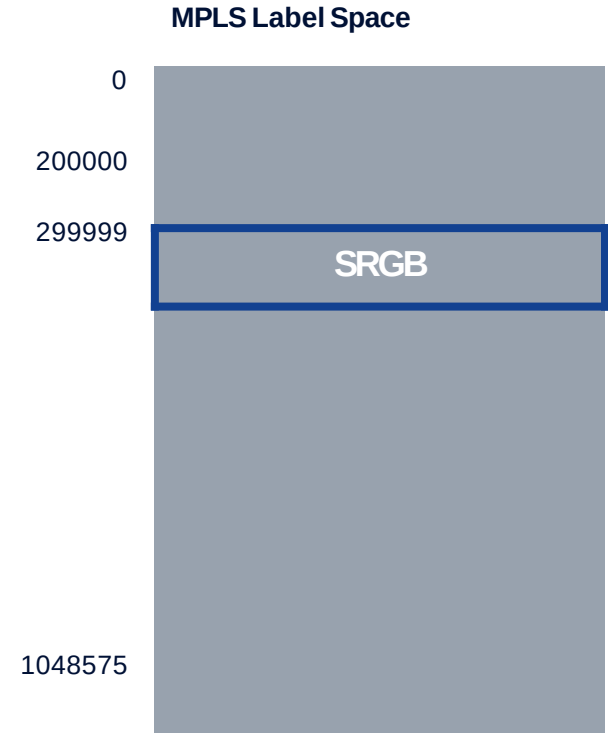
Segment routing με MPLS data plane (2)

- Όταν γίνεται χρήση του MPLS data-plane για το segment routing, τότε :
 - Η λίστα των segments αναπαρίσταται σαν μια στοίβα (stack) labels
 - Το active segment είναι στην κορυφή της στοίβας (top/head label)
 - Η λειτουργία **PUSH** υλοποιείται σαν **PUSH (MPLS) (νέο TOP με προσθήκη)**
 - Η λειτουργία **NEXT** υλοποιείται σαν **POP (MPLS) (νέο TOP με αφαίρεση)**
 - Η λειτουργία **CONTINUE** υλοποιείται σαν **SWAP** (δεν τροποποιεί το βάθος της λίστας)



Segment routing με MPLS data plane (3)

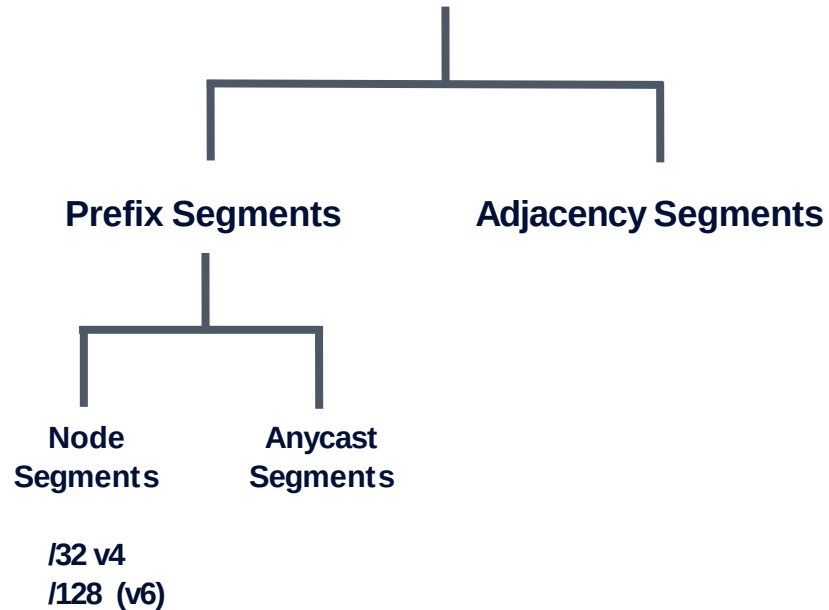
- Segment Routing Global Block (SRGB)
 - SRGB ένα σύνολο από τοπικά (local labels) δεσμευμένα για global segments
 - Τοπική ιδιότητα ενός SRnode
 - Με ίδια SRGB σε όλους τους κόμβους σε ένα SR domain απλουστεύεται η αρχικοποίηση της ανάθεσης SiD



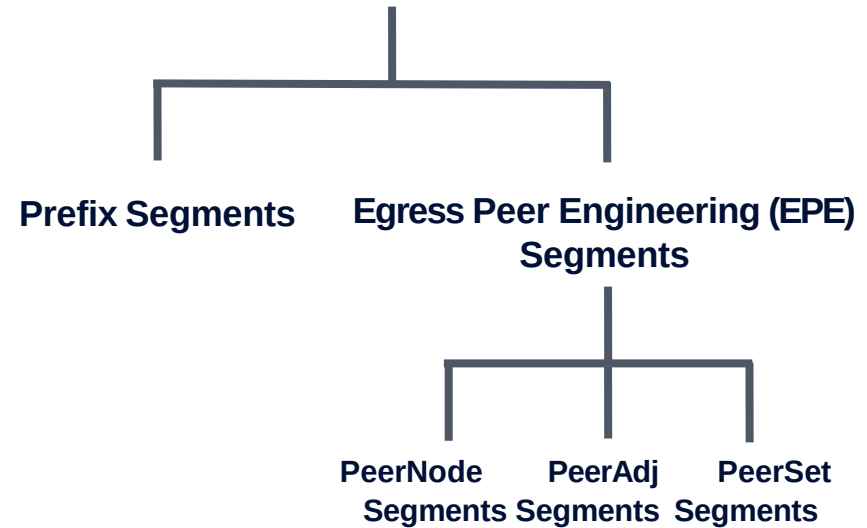
Είδη segments



IGP SEGMENTS



BGP SEGMENTS



Είδη segments

Prefix Segment

- Μοναδικό από την SRGB
- Τυπικά multi-hop
- ECMP-aware shortest- path IGP router σε ένα prefix
- Indexed ή absolute SID
- Σηματοδοτούνται από το IGP

Adjacency Segment

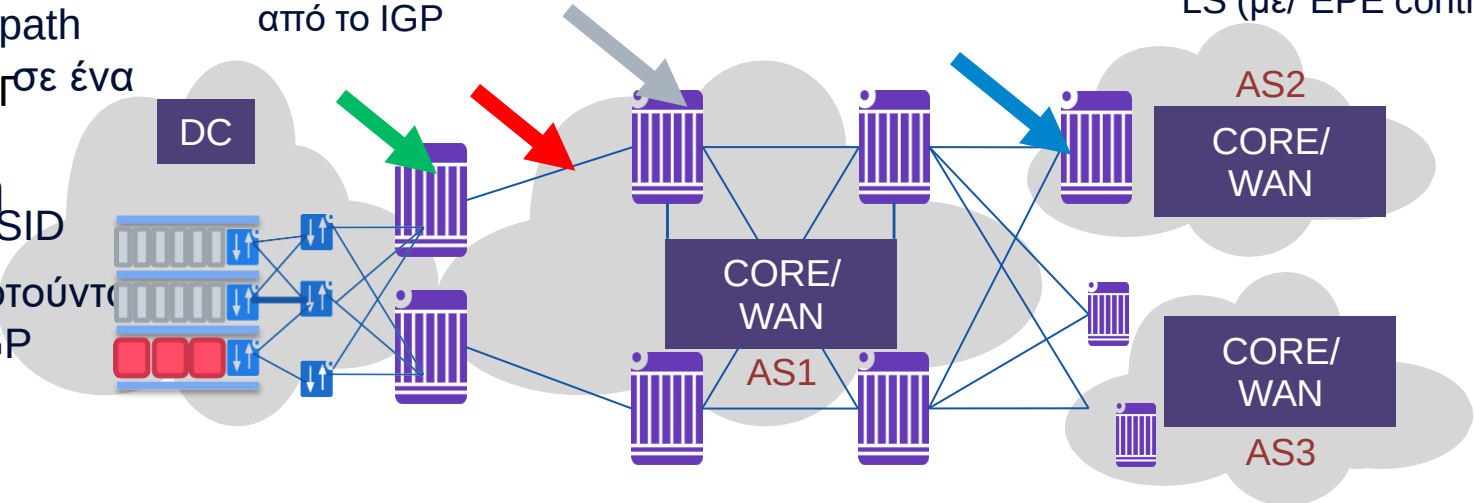
- Τοπικά μοναδικό – κάθε SR router στο domain μπορεί να χρησιμοποιήσει τον ίδιο χώρο
- Typically single-hop
- Σηματοδοτούνται από το IGP

BGP Prefix Segment

- Π.χ. Prefix Segment στο DC
- DC GW representation
- Σηματοδοτούνται από BGP (στο DC)

BGP Peer Segment

- EPE ; Egress Peering Engineering
- Επηρεασμός της κίνησης προς το (→) **ΓΕΙΤΟΝΙΚΟ AS**
- Σηματοδοσία από BGP-LS (με/ EPE controller)





Segment identifiers – prefix segments

- Prefix Segment (Prefix-SID)
 - Μοναδικό εντός IGP/SR – από το SR Global Block (SRGB)*
 - Αντιστοιχεί σε ένα prefix με χρήση ενός ECMP shortest-path IGP route προς το prefix
 - Τυπικά ένα multi-hop path
 - Περιέχει ένα “P” flag για να επιτρέπει στους γείτονες να κάνουν λειτουργία “NEXT” (pop) ενώ επεξεργάζονται το segment (αντίστοιχα με το Penultimate Hop Popping στο MPLS).
 - Δύο παραλλαγές; Indexed ή Absolute-SID (σε επόμενα slides)

* Σε περιβάλλον MPLS, SRGB: σύνολο των local labels δεσμευμένα για global segments.

Segment identifiers – node segments

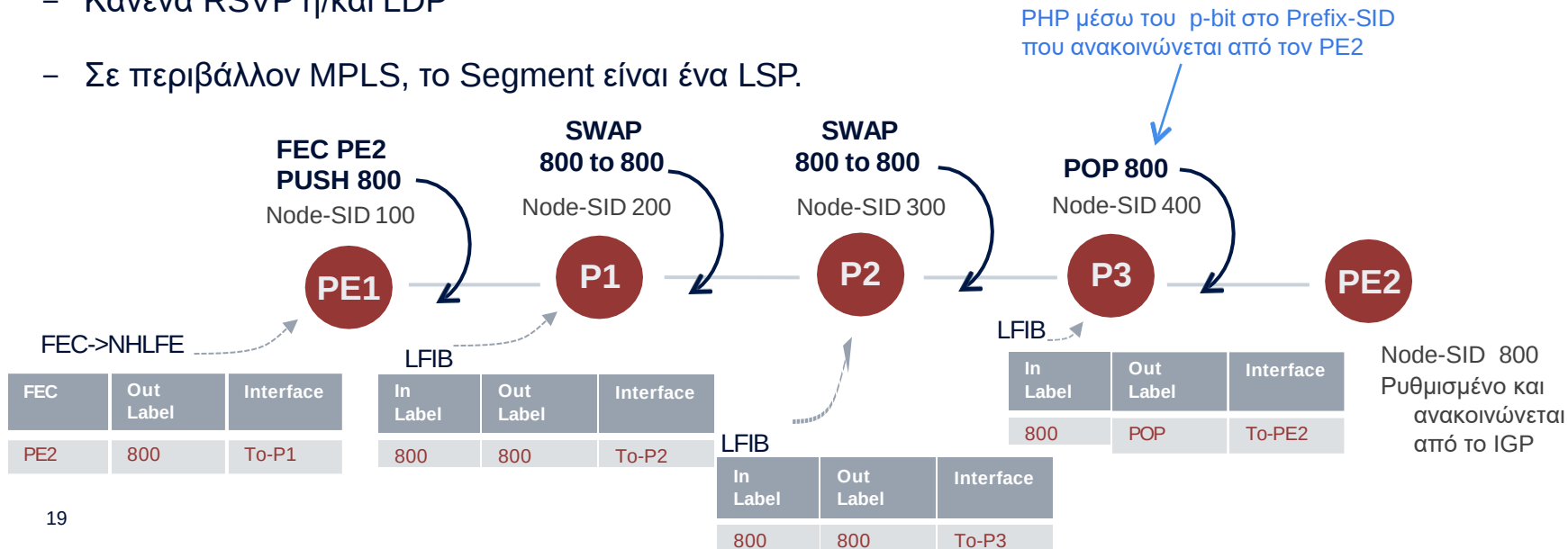
- Node Segment ID (Node-SID)
 - Ένα ειδικό prefix segment που περιγράφει ένα router (loopback/system address).
 - Ταυτοποιείται από το “N” flag στα ανακοινωμένα segment (Prefix-SID Sub-TLV).
 - Αναπαριστά τις διαδρομές ECMP με χρήση των shortest-path IGP route προς το συγκεκριμένο κόμβο.

Segment identifiers – anycast segments

- Anycast Segment ID (Anycast-SID)
 - Ένα prefix segment για ένα σύνολο routers (π.χ. SPINES σε ένα DC)
 - Αναπαριστά μια διαδρομή (ECMP-aware shortest-path IGP) στο πλησιέστερο κόμβο του “anycast set”.
 - Δυνητικά χρήσιμο για χοντρικού τύπου traffic engineering. (π.χ. φτάσε σε ένα SPINE router)

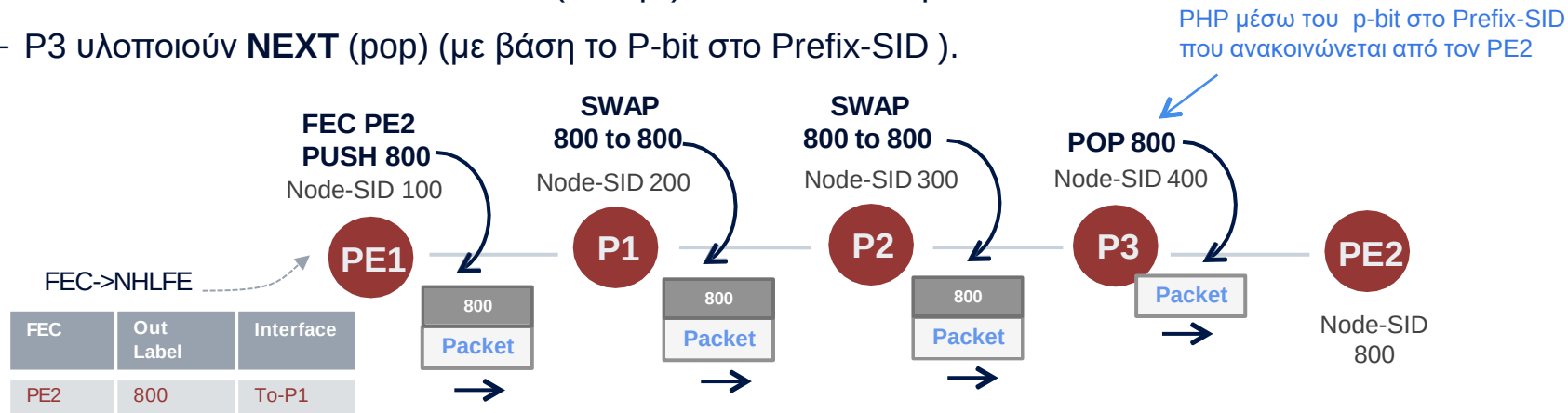
Παράδειγμα: SR tunnel με prefix-SID (node-SID)

- PE2 ανακοινώνει Node Segment στο IGP (Prefix-SID Sub-TLV Extension στο IS-IS/OSPF)
- Όλοι οι δρομολογητές σε ένα SR domain εγκαθιστούν node segment προς το PE2 με MPLS data-plane (LFIB iLM).
- Κανένα RSVP ή/και LDP
- Σε περιβάλλον MPLS, το Segment είναι ένα LSP.



Παράδειγμα : SR tunnel με prefix-SID (node-SID)

- Για κίνηση από PE1 → PE2, PE1 **PUSH** στο node segment {800} και χρήση shortest IGP path για να φτάσει στο PE2.
- Το Active segment είναι στην κορυφή της στοίβας σε περιβάλλον MPLS:
 - P1 και P2 υλοποιούν **CONTINUE** ("swap") σε MPLS data-plane
 - P3 υλοποιούν **NEXT** (pop) (με βάση το P-bit στο Prefix-SID).



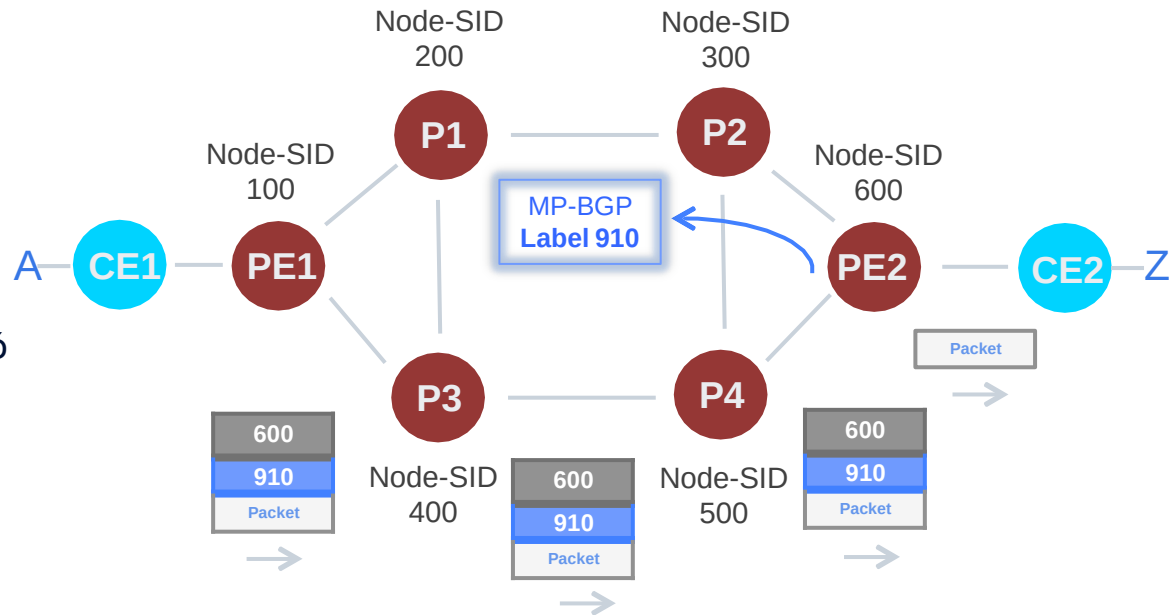
- Καμία πληροφορία (state)/path στο δίκτυο με εξαίρεση την λίστα των segment για την αρχή του tunnel που συντηρείται στον PE1.

Prefix segment identifiers – absolute SIDs

- Η χρήση absolute SID απαιτεί μια μοναδική SRGB σε όλους τους SR routers σε όλο το IGP domain.

- Παράδειγμα:

- PE2 ανακοινώνει service Label MP-BGP (RFC4364) label 910 για το IP-VPN prefix Z.
- Κίνηση -> VPN prefix Z, και υποθέτοντας (non-ECMP) path από PE1 -> PE2 είναι PE1-P3-P4- PE2, Το PE1 ΣΠΡΩΧΝΕΙ label 910 στο κάτω μέρος της στοίβας, και label 600 (Node-SID για PE2) στην κορυφή της στοίβας.



- Label (SID) δεν αλλάζει από hop σε hop

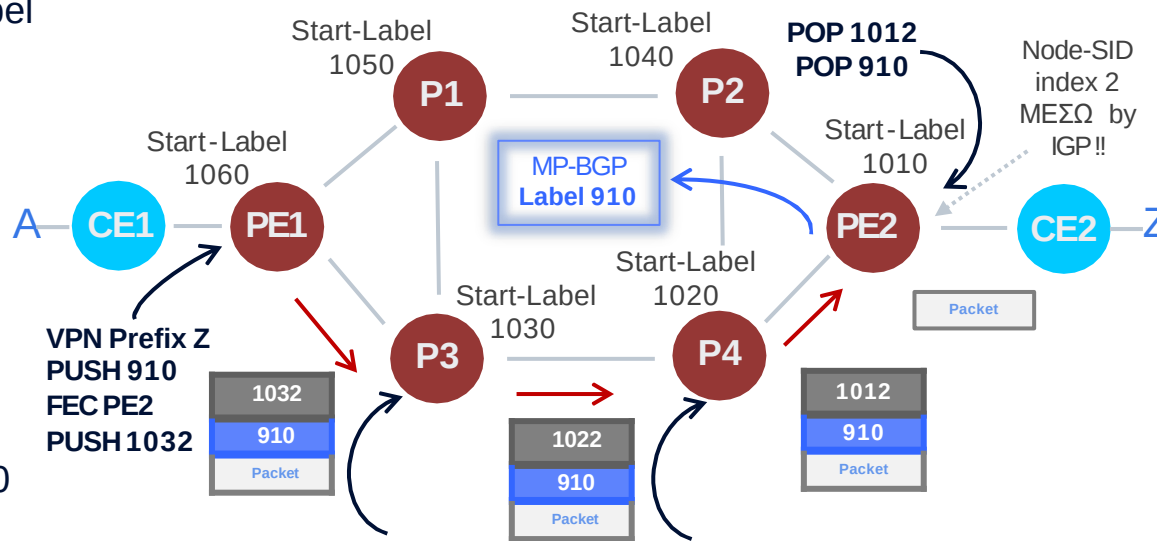


Prefix SID indexing

- Λόγος ?
 - Ένα SR domain μπορεί να περιέχει πολλαπλούς κατασκευαστές με την πιθανότητα κάθε κατασκευαστής να έχει το δικό του εύρος από MPLS label
 - Το Prefix SID πρέπει να μοναδικό στο SR domain (Αντί για global unique SRGB μοναδικό index)
- Πως ?
 - Ο μηχανισμός Indexing χρειάζεται για αναζήτηση των prefix SIDs. Όλοι οι δρομολογητές σε ένα SR domain πρέπει να ρυθμιστούν και να ανακοινώνουν το **ΙΔΙΟ** prefix SID index range για κάθε διαφορετικό IGP.
 - Η τιμή του label που χρησιμοποιείται από κάθε router για ένα prefix 'Z' (στο LFIB = label στο ILM) μπορεί να είναι τοπική στο router με χρήση ενός offset label, το λεγόμενο και start label :
Τοπικά Label (για Prefix SID) = (τοπικό) start-label + {Prefix SID index}

Παράδειγμα: Prefix Segments αλλά με SID indexing

- Παράδειγμα, έστω SID Index Range είναι {1,100}.
- Κάθε SR router στο domain ορίζει το αρχικό label στην SRGB (start-label), ένα offset label called an SID index.
 - SR routers αθροίζουν {start-label + SID index} για να φτιάξουν ένα local label για το Prefix SID.
- Έστω PE2 ανακοινώνει το loopback 192.0.2.2/32 με prefix index to 2:
 - PE2's SID είναι $\{1010+2\}= 1012$
 - P4's SID είναι $\{1020+2\}= 1022$
 - P3's SID είναι $\{1030+2\}=1032$ κοκ.
 - PE2 ανακοινώνει MP-BGP label 910 για το VPN prefix Z.



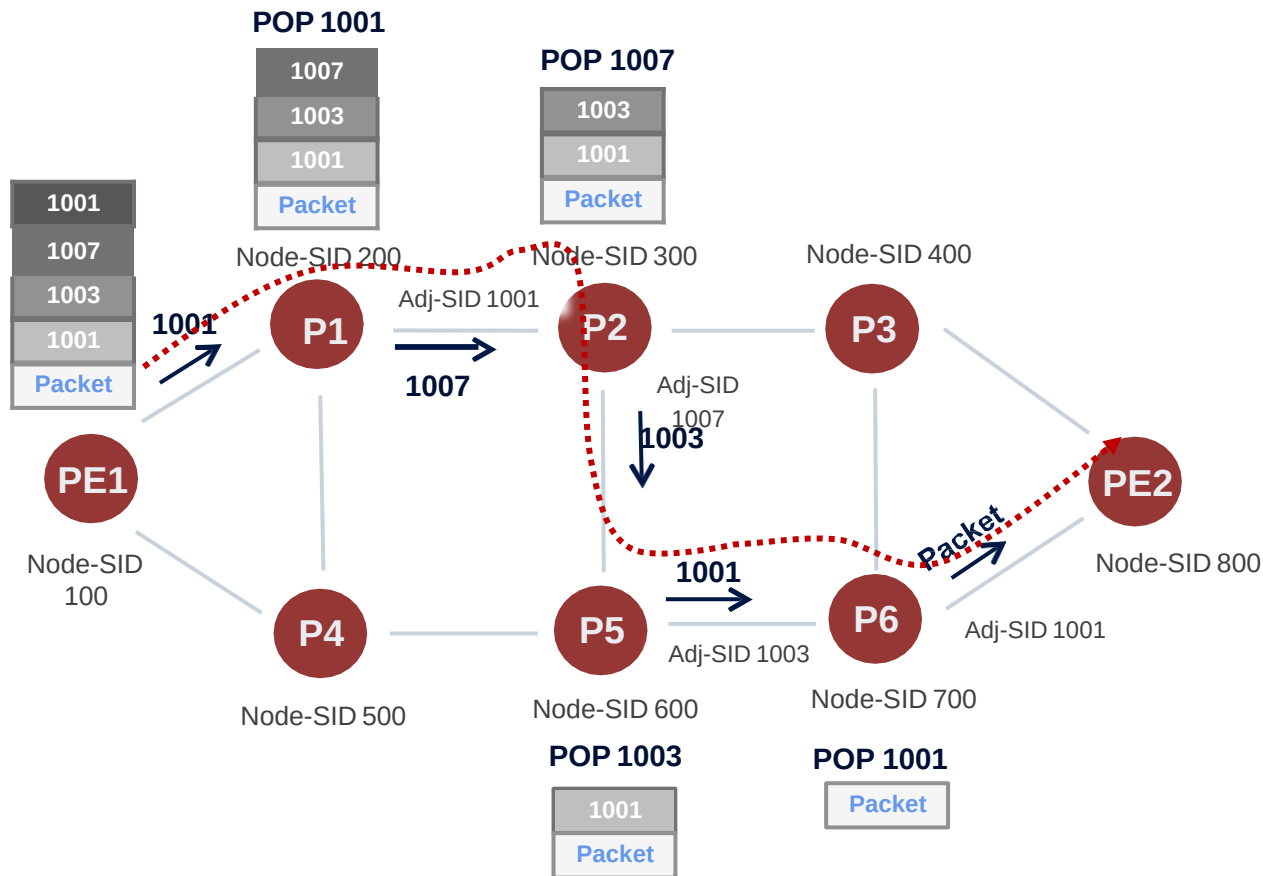
Segment identifiers – adjacency segments

- Όλοι οι SR routers ανακοινώνουν Adjacency segment(s) στο IGP (Adjacency-SID Sub- TLV Extension στο IS-IS/OSPF).
- Τα Adjacency segments είναι τοπικά, αλλά μόνο ο ανακοινών SR router εγκαθιστά το adjacency segment στο MPLS data-plane
 - *Κατά αντιστοιχία με data-path, είναι αντίστοιχο του label-swap στο implicit-null.*
- Κατάλληλο για από άκρη-σε-άκρη (end-to-end) source-routing όπου τα Adjacency segments καθορίζουν επακριβώς την διαδρομή hop-by-hop διαμέσου του δικτύου.
- Προσοχή, το βάθος του label stack έχει επιπτώσεις στο χειρισμό του από το υλικό του δρομολογητή.

Segment identifiers – adjacency segments

- Adjacency Segment ID (Adj-SID)
 - Ένα segment που καθορίζει ένα link (adjacency) ή ένα σύνολο τους που είναι στο IGP.
 - Το Segment Identifier (SID) ανακοινώνεται (από κάθε SR router στο SR domain).
 - Εάν:
 - AB είναι το Node-SID του node N, και...
 - ABC είναι το Adj-SID στο node N για ένα adjacency πάνω από το link L, τότε....
 - Ένα πακέτο με segment list {AB, ABC} θα προωθηθεί με shortest-path -> node N, με μεταγωγή από N προς link L, **ΧΩΡΙΣ** shortest-path routing.
 - Εάν Adj-SID ταυτοποιεί ένα σύνολο από adjacencies, node N μπορεί να μοιράσει (load-balance) στα μέλη του συνόλου.

Παράδειγμα: SR tunnel με adjacency segments

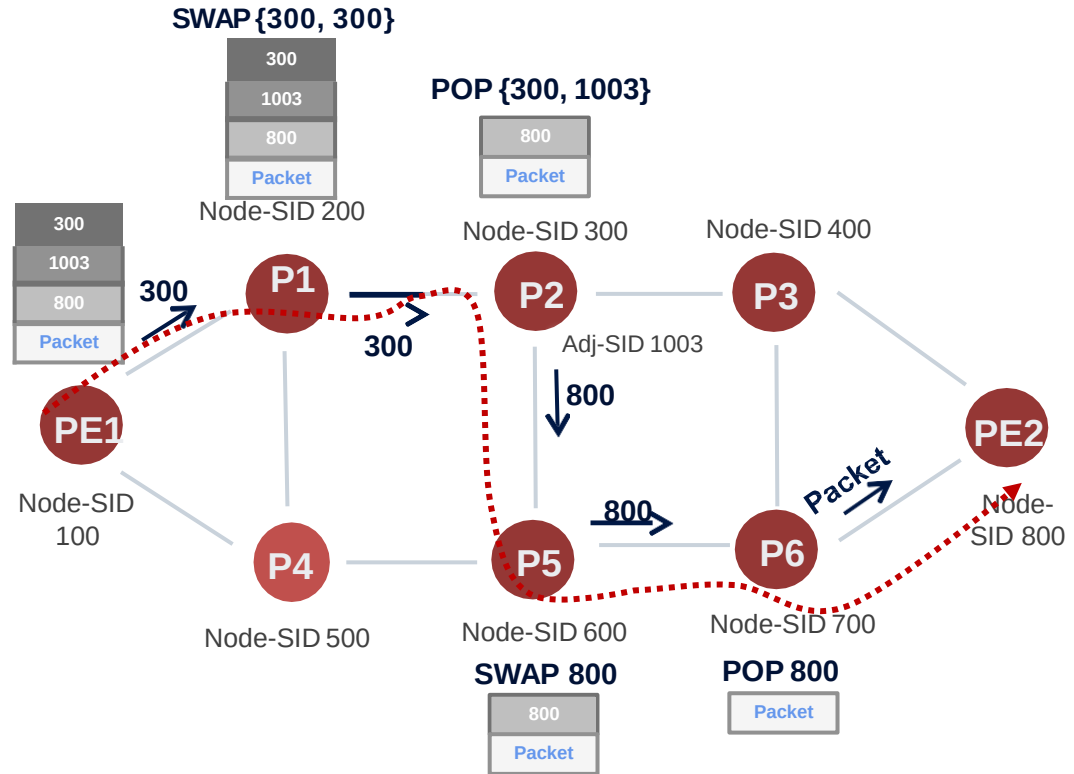


Παράδειγμα: SR tunnel με node και adjacency segments

- Ο συνδυασμός node και adjacency segments είναι εφικτός.
- Αυτό ο συνδυασμός προσφέρει την δυνατότητα για ECMP paths προς τον επόμενο node segment, αλλά με χρήση ενός (ή συνόλου links) από αυτό το κόμβο.

Παράδειγμα: SR tunnel με node και adjacency segments

- PE1 θέλει να διασχίσει το link P2-P5 καθοδόν προς το PE2, επειδή είναι υποχρησιμοποιούμενο.
- PE1 προσθέτει το segment list {300, 1003, 800} αναπαριστώντας το Node-SID για το P2, Το Adj-SID για το link P2-P5, και τέλος το Node-SID για το PE2.



Σύγκριση LDP <-> RSVP-TE<-> SR

	LDP	RSVP-TE	SR
Overview	Multipoint to point	Point to point	Multipoint to point
Operation	Simple	LSP per destination/TE- path	Simple
Dependencies	Relies on IGP	Relies on IGP TE	Relies on IGP + offline TE
LBL allocation	Local significant per node (interface)	Local significant per node (interface)	Global
Traffic Engineering	No	Yes	yes
Scaling	1 LBL per node (interface)	Nx(N-1)	1 LBL per node/ local interface
Fast Reroute	LFA, LFA Policies, RLFA - <100% coverage	Link/Node protection (detour/facility) – 100% coverage	LFA, LFA Policies, RLFA/DLFA - can get to 100% coverage (better than LDP with RLFA)
Multicast	mLDP	P2MP RSVP	TBD
IPv6	Extensions required	Extensions required	Native

SRV6

IPv6 Forwarding

SRv6 Μηχανισμοί



Encapsulating mode

- SR ingress router κάνει ENCAPSULATION payload packet σε ένα IPv6 header
- Source node ενσωματώνει ένα routing extension header μεταξύ IPv6 header και payload (.. specs)

Simplified mode

SR ingress προσθέτει ένα routing extension header μεταξύ του payload IPv6 header και payload data

(πρώτες υλοποιήσεις- πραγματιστική προσέγγιση)

Και στις δύο περιπτώσεις:

To routing extension header μεταφέρει την στοίβα των SIDs

SRv6 Μηχανισμοί



Transit behaviours

IPv6

IN-MAC header	IPv6 header	Transport header	Payload
---------------	-------------	------------------	---------

T.Insert

OUT-MAC header	IPv6 header	SRH	Transport header	Payload
----------------	-------------	-----	------------------	---------

T.Encaps

OUT-MAC header	IPv6 header	SRH	IPv6 header	Transport header	Payload
----------------	-------------	-----	-------------	------------------	---------

T.Encaps.L2

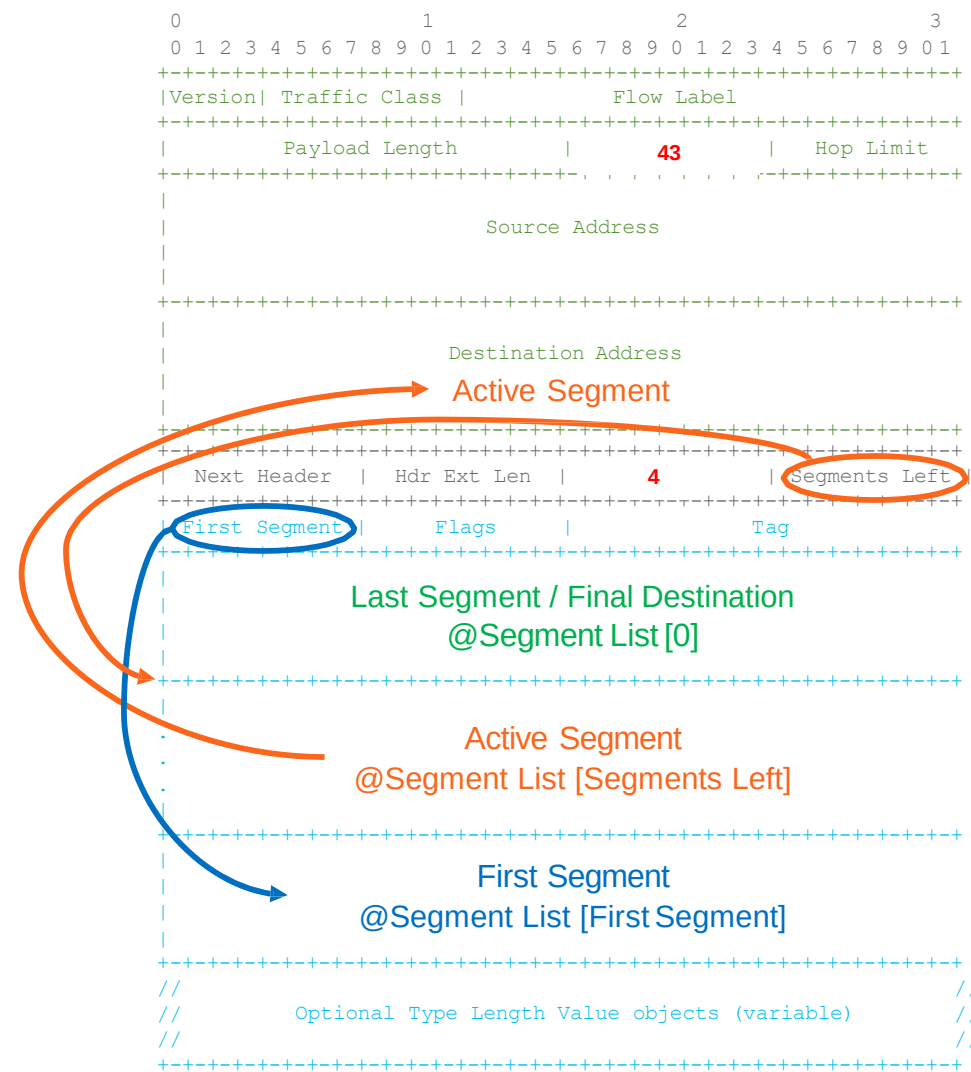
OUT-MAC header	IPv6 header	SRH	IN-MAC header	IPv6 header	Transport header	Payload
----------------	-------------	-----	---------------	-------------	------------------	---------

Ορισμός/Τυποποίηση Λειτουργιών:

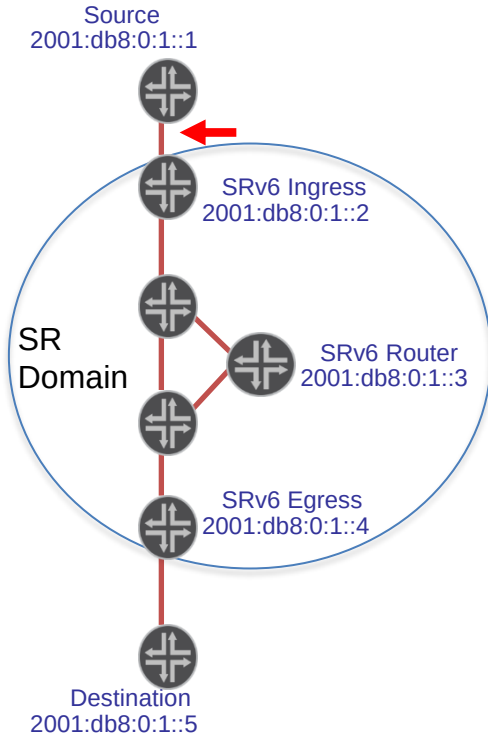
Codename	Behavior	
End	Endpoint	[PSP/USP flavors]
End.X	Endpoint with Layer-3 cross-connect	[PSP/USP flavors]
End.B6	Endpoint bound to an SRv6 policy	
End.B6.Encaps	Endpoint bound to an SRv6 Encapsulation policy	
End.DX6	Endpoint with decapsulation and IPv6 cross-connect	(per-CE VPN label)
End.DX4	Endpoint with decapsulation and IPv4 cross-connect	(per-CE VPN label)
End.DT6	Endpoint with decapsulation and specific IPv6 table lookup	(per-VRF VPN label)
End.DT4	Endpoint with decapsulation and specific IPv4 table lookup	(per-VRF VPN label)
End.DX2	Endpoint with decapsulation and Layer-2 cross-connect	

IPv6 SR Header

- Κάθε segment είναι IPv6 δ/νση
- Τα Segments κωδικοποιούνται ΑΝΑΣΤΡΟΦΑ
- ΤΕΛΕΥΤΑΙΟ segment index είναι 0
- Πρώτο segment index είναι First Segment
- Active segment index είναι Segments Left
- **Active Segment ΑΝΤΙΓΡΑΦΕΤΑΙ ΣΤΟ Destination Address πεδίο του IP header**



Segment Routing Header (SRH) : (1 of 6)



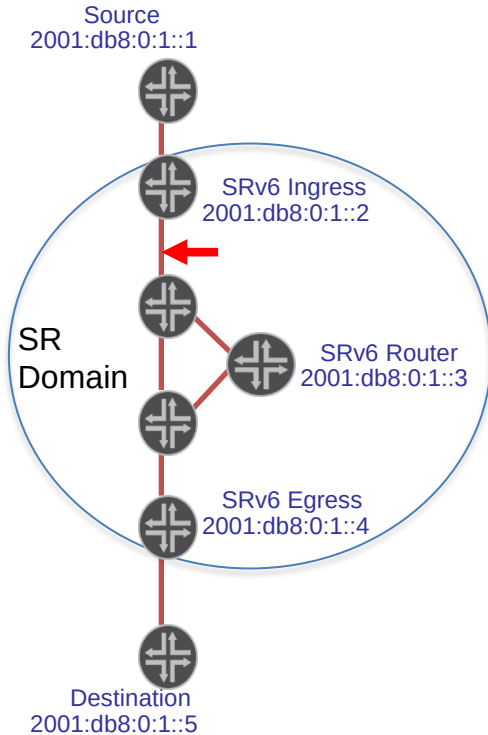
Ver	DSCP	Flow Label	
Payload Length 40		Next HDR TCP	Hop Limit 255
Source Address 2001:db8:0:1::1			
Destination Address 2001:db8:0:1::5			
TCP Header and Payload			

IPv6
HEADER

TCP
Header/Payload



Segment Routing Header (SRH) : (2 of 6)



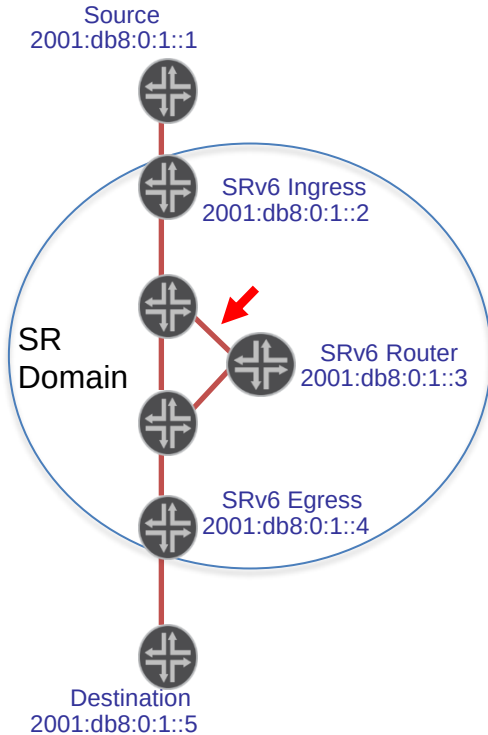
Ver	DSCP	Flow Label	
Payload Length 136		Next HDR SRH	Hop Limit 255
Source Address 2001:db8:0:1::2			
Destination Address 2001:db8:0:1::3			
Next HDR TCP	Length 56	HDR Type 4	Seg Left 1
Last Entry 2	Flags		Tag
Segment 0 2001:db8:0:1::4			
Segment 1 2001:db8:0:1::3			
Segment 2 2001:db8:0:1::2			
Ver	DSCP	Flow Label	
Payload Length 40		Next HDR TCP	Hop Limit 255
Source Address 2001:db8:0:1::1			
Destination Address 2001:db8:0:1::5			
TCP Headed and Payload			

IPv6
HEADER

Segment
Routing
Extension
Header

Payload
(TCP/IPv6)

Segment Routing Header (SRH) : (3 of 6)



Ver	DSCP	Flow Label	
Payload Length 136		Next HDR SRH	Hop Limit 254
Source Address 2001:db8:0:1::2			
Destination Address 2001:db8:0:1::3			
Next HDR TCP	Length 56	HDR Type 4	Seg Left 1
Last Entry 2	Flags		Tag
Segment 0 2001:db8:0:1::4			
Segment 1 2001:db8:0:1::3			
Segment 2 2001:db8:0:1::2			
Ver	DSCP	Flow Label	
Payload Length 40		Next HDR TCP	Hop Limit 255
Source Address 2001:db8:0:1::1			
Destination Address 2001:db8:0:1::5			
TCP Headed and Payload			

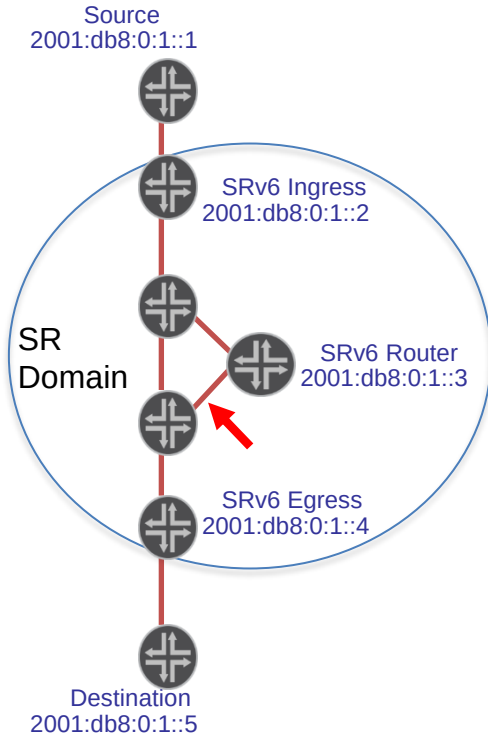
IPv6
HEADER

Segment
Routing
Extension
Header

Payload
(TCP/IPv6)



Segment Routing Header (SRH) : (4 of 6)



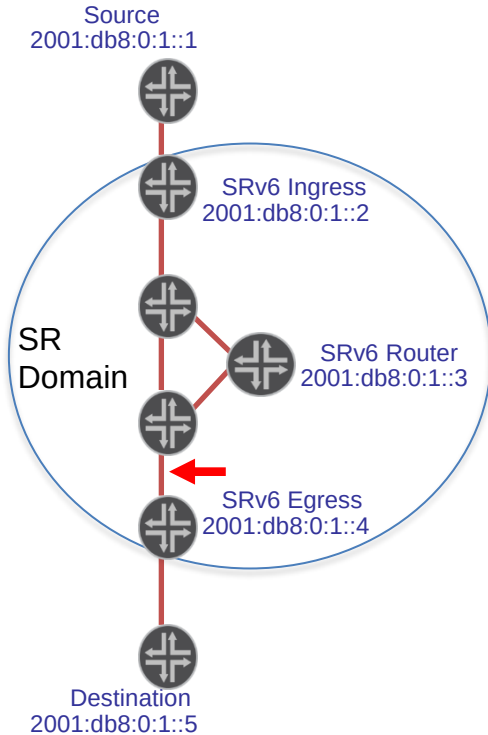
Ver	DSCP	Flow Label	
Payload Length 136		Next HDR SRH	Hop Limit 253
Source Address 2001:db8:0:1::2			
Destination Address 2001:db8:0:1::4			
Next HDR TCP	Length 56	HDR Type 4	Seg Left 0
Last Entry 2	Flags		Tag
Segment 0 2001:db8:0:1::4			
Segment 1 2001:db8:0:1::3			
Segment 2 2001:db8:0:1::2			
Ver	DSCP	Flow Label	
Payload Length 40		Next HDR TCP	Hop Limit 255
Source Address 2001:db8:0:1::1			
Destination Address 2001:db8:0:1::5			
TCP Headed and Payload			

IPv6
HEADER

Segment
Routing
Extension
Header

Payload
(TCP/IPv6)

Segment Routing Header (SRH) : (5 of 6)



Ver	DSCP	Flow Label	
Payload Length 136		Next HDR SRH	Hop Limit 252
Source Address 2001:db8:0:1::2			
Destination Address 2001:db8:0:1::4			
Next HDR TCP	Length 56	HDR Type 4	Seg Left 0
Last Entry 2	Flags		Tag
Segment 0 2001:db8:0:1::4			
Segment 1 2001:db8:0:1::3			
Segment 2 2001:db8:0:1::2			
Ver	DSCP	Flow Label	
Payload Length 40		Next HDR TCP	Hop Limit 255
Source Address 2001:db8:0:1::1			
Destination Address 2001:db8:0:1::5			
TCP Headed and Payload			

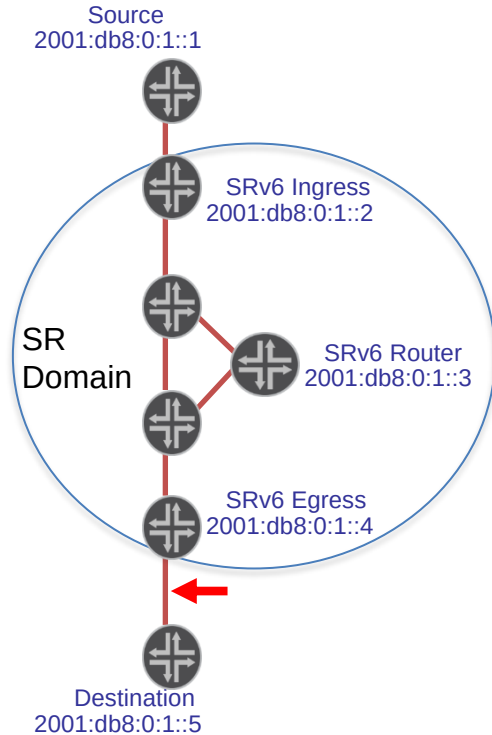
IPv6
HEADER

Segment
Routing
Extension
Header

Payload
(TCP/IPv6)



Segment Routing Header (SRH) : (6 of 6)

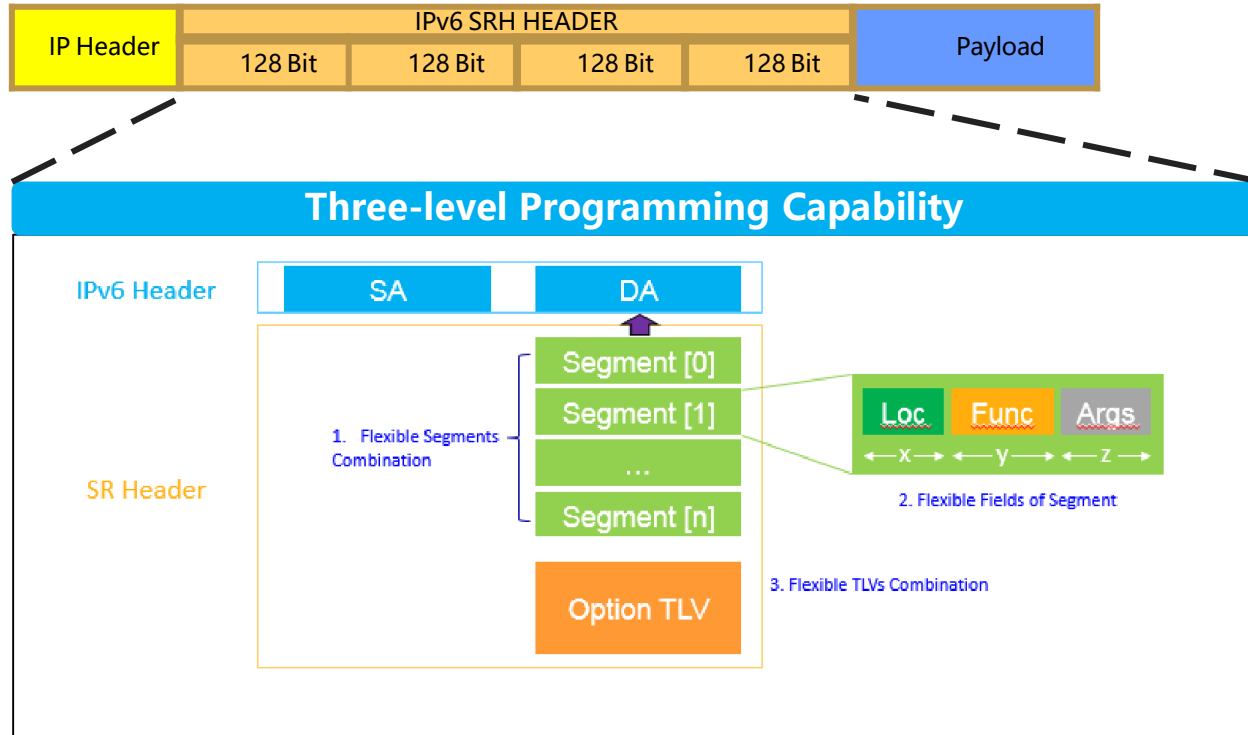


Ver	DSCP	Flow Label	
Payload Length 40		Next HDR TCP	Hop Limit 254
Source Address 2001:db8:0:1::1			
Destination Address 2001:db8:0:1::5			
TCP Header and Payload			

IPv6
HEADER

TCP
Header/Payload

SRv6: Network Programming



SRv6 network programming innovation

Segment format



Locator

Function

1111 : 2222 : 3333 : 4444 : 5555 : 6666 : 7777 : 8888

- SRv6 SIDs : 128-bit addresses
 - Locator: Τα πιο σημαντικά για δρομολόγηση του segment
 - Function: Τα λιγότερα σημαντικά για λειτουργίες από το node
- Flexible bit-length allocation
 - Segment format έχουν τοπική σημασία για κάθε SR κόμβο
 - Θυμίζει λίγο LISP πρωτόκολλο



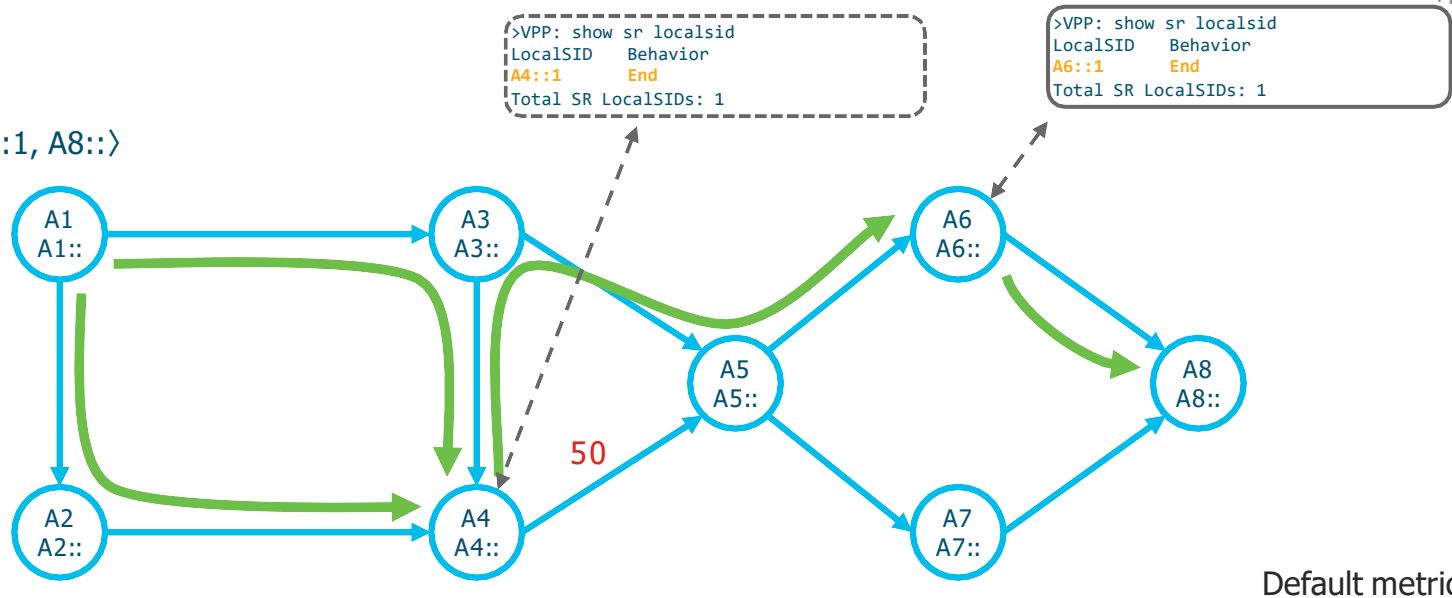
SRv6

LocalSIDs



Endpoint function

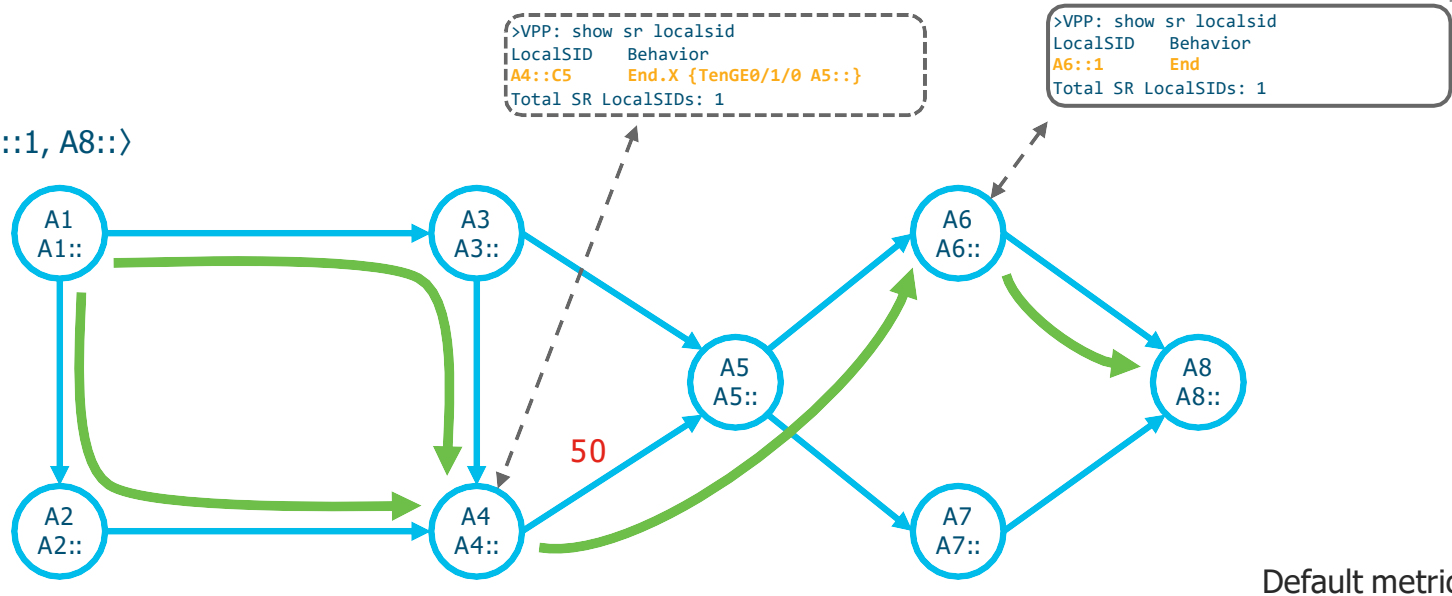
SR: A4::1, A6::1, A8::>



- Για απλότητα function 1 σημαίνει την βασική λειτουργία (function)
- Shortest-path → Node

Endpoint και xconnect → neighbor function

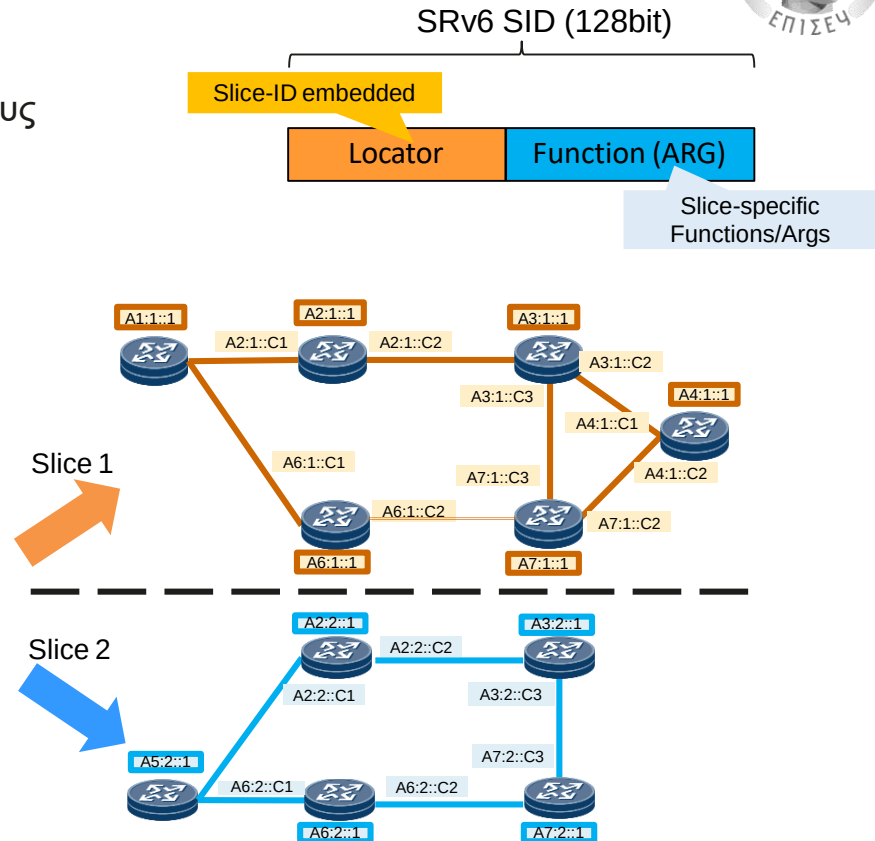
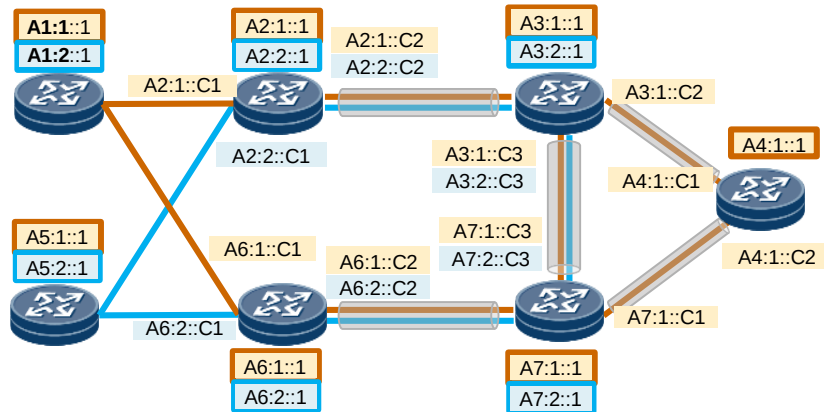
SR: A4::C5, A6::1, A8::>



- For simplicity $A_k::C_j$ denotes:
 - Shortest-path to the Node K and then x-connect (function C) to the neighbor J

SRv6 για Network Slicing: Data Plane

- SRv6 ειδικά για network resource awareness
- Τα τοπικά SRv6 SIDs αναπαριστούν δικτυακούς πόρους στους οποίους δρουν
- Ταυτοποίηση του Network slice identification
 - Ανάθεση SRv6 Locator για κάθε network slice
 - SRv6 SIDs κληρονομούν το slice από τον Locator



Segment format -uSID

BBBB: BBBB:uSID1:uSID2: uSID3: uSID4 :uSID5 :uSID6

BBBB:BBBB:<uSID1>:<uSID2>:<uSID3>:<uSID4>:<uSID5>:<uSID6>

Function: το SID είναι πλέον Container SID με αλληλουχία από μικρο SID

Μειώνεται το stack των SRH

Allocation από Private IPv6 space

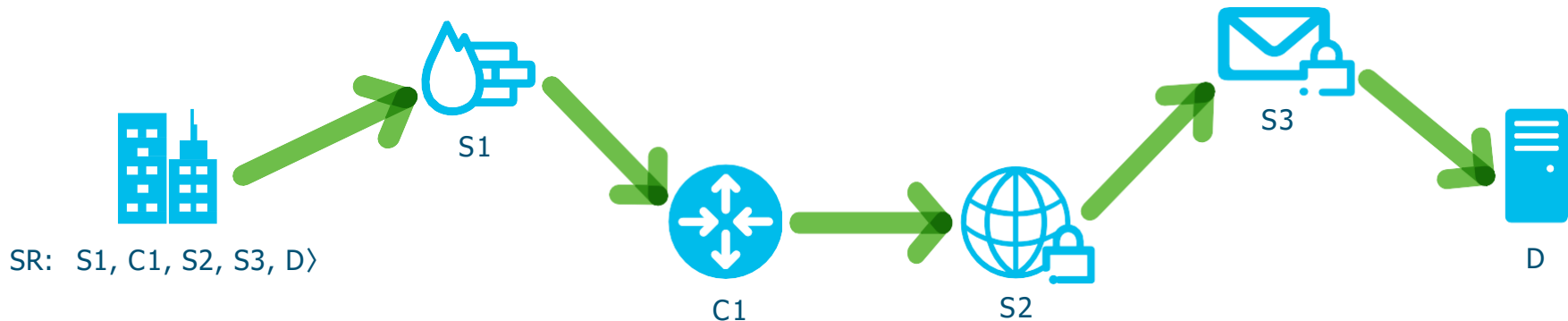
Ανάθεση των micro SID σε τερματικούς κόμβους , προγράμματα ή και docker containers

Δοκιμάστηκαν σε Alibaba k Bell Canada

Υλοποίηση σε Cisco 8000, Cisco ASR9000, Cisco NCS5500, Cisco XRv9K series, Barefoot Tofino, Arrcus με Jericho2, Marvell Falcon, Linux kernel and FD.IO VPP.

Service Chaining με SRv6

Packets καθοδηγούνται μέσω ακολουθίας από services/function/Υπηρεσίες → server



- Οι Υπηρεσίες ~ ως segments

- Ευελιξία
- Επεκτάσιμα
- Κανένα State



Εφαρμογές



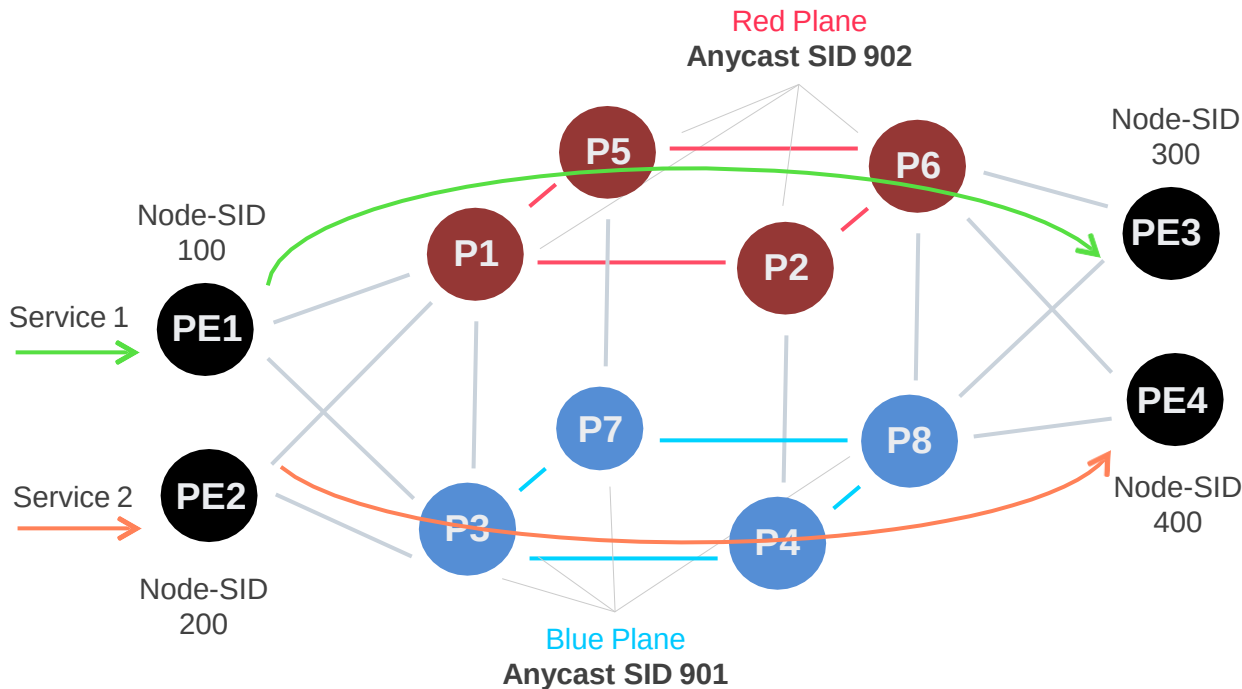
Use case 3: Disjointness (1)

- Δύο υπηρεσίες χωρίς κοινές διαδρομές. – π.χ. βλάβη στη μία δεν επηρεάζει την άλλη.
- Disjointness με χρήση Anycast segments.

Περίπτωση 1: Disjointness (2)

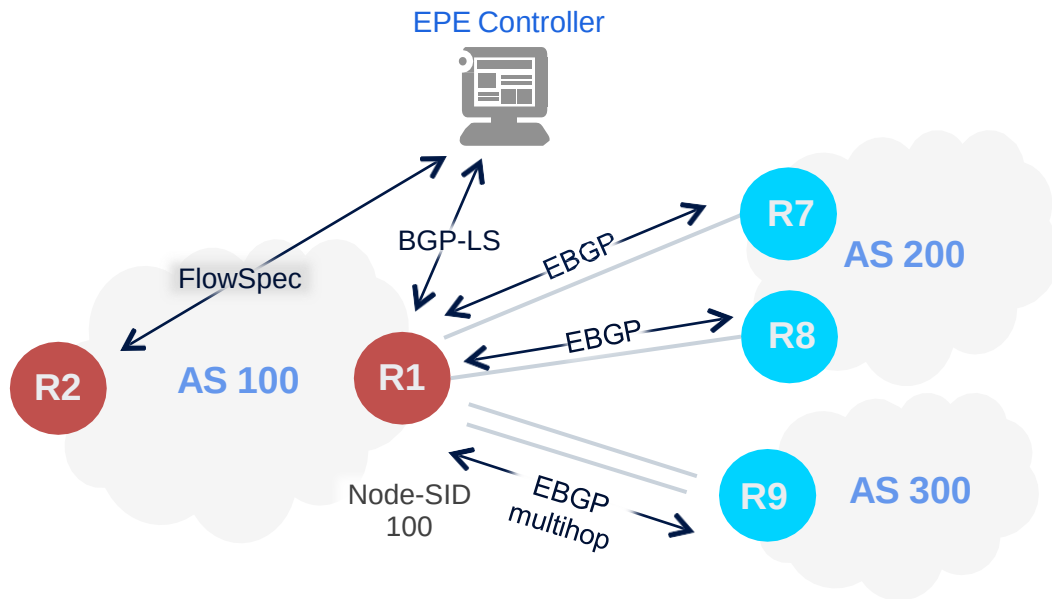
- Έστω service 1 μεταξύ PE1 και PE3 να μην περνά από service 2 μεταξύ PE2 κ PE4:

- Service 1 στο PE1 έχει segment list {902, 300} συμπεριλαμβανομένου του Anycast SID 902 που διασχίζει το **κόκκινο** plane πριν φτάσει στο PE3.
- Service 2 στο PE2 έχει segment list {901, 400} συμπεριλαμβανομένου του Anycast SID 901 που διασχίζει το **μπλε** προτού φτάσει PE4.



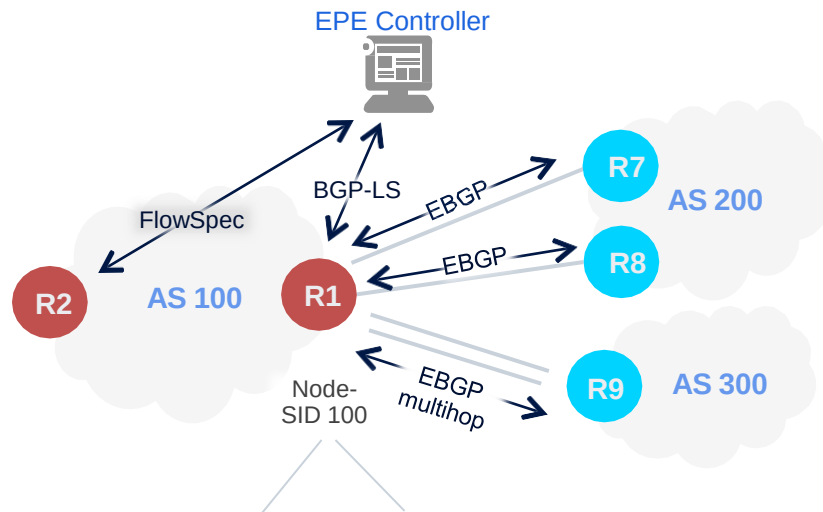
Περίπτωση 2: Egress peer engineering (EPE) (1)

- Egress Peer Engineering ορίζει τρία BGP Peering SIDs, για προγραμματισμό source-routed inter-domain paths; PeerNodeSID, PeerAdjSID, και PeerSetSID.
- R1 είναι ένας EPE egress router που δεσμεύει τα ακόλουθα:
 - PeerNode segment για καθένα από τους peers (R7, R8, και R9)
 - PeerAdj segment για κάθε recursive interface σε ένα multi-hop peer (R9)
 - PeerSet segment σε ένα σύνολο από peer (R7 και R8) (AS200)



Use case 3: Egress peer engineering (EPE) (2)

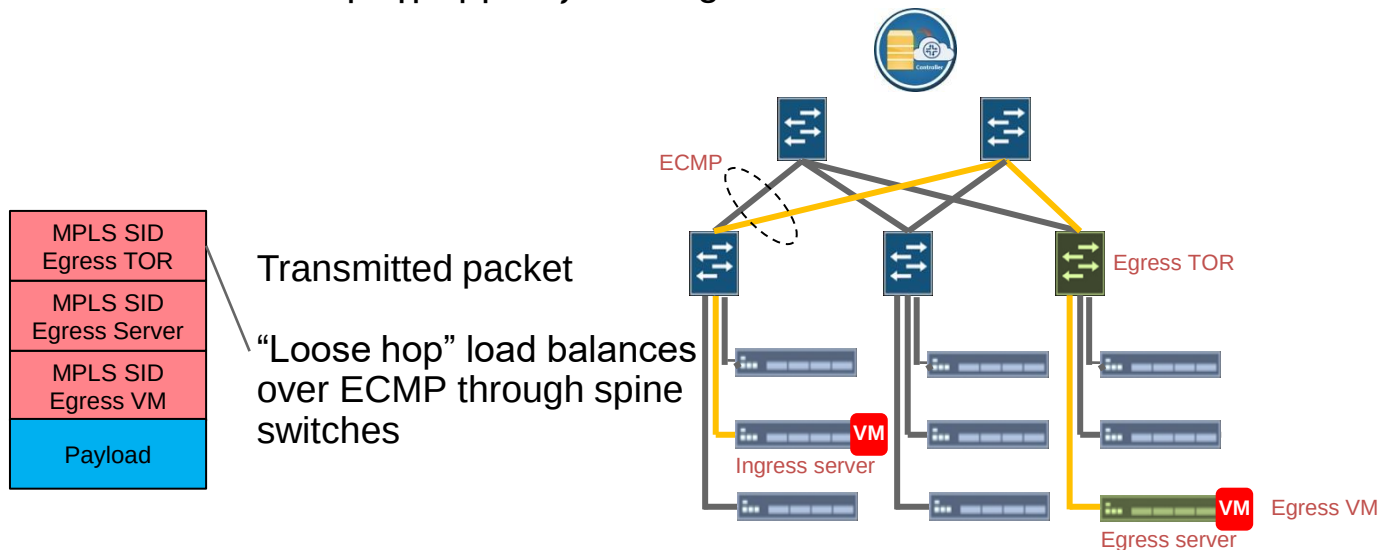
- BGP-LS session μεταξύ EPE-border router (R1) και EPE controller:
 - R1 ανακοινώνει PeerNode, PeerAdj, και PeerSet SIDs με SR extensions στο BGP-LS, και ανάλογα προο/ζει το FIB.
- EPE Controller προο/ζει source-routes από τον ingress routers προς E-BGP peers με FlowSpec
 - 80% traffic -> AS 300 με segment list {100, 1005}
 - 20% traffic -> AS 200 με segment list {100, 1006}
 - Prefix <NLRI/Length> segment list {100, 1003}
 - Prefix <NLRI/Length> segment list {100, 1004}



Incoming Label	Operation	Outgoing Interface
1001	POP	Link to R7
1002	POP	Link to R8
1003	POP	Upper link to R9
1004	POP	Lower link to R9
1005	POP	Load-balance on any link to R9
1006	POP	Load-balance on any link to R7 or R8

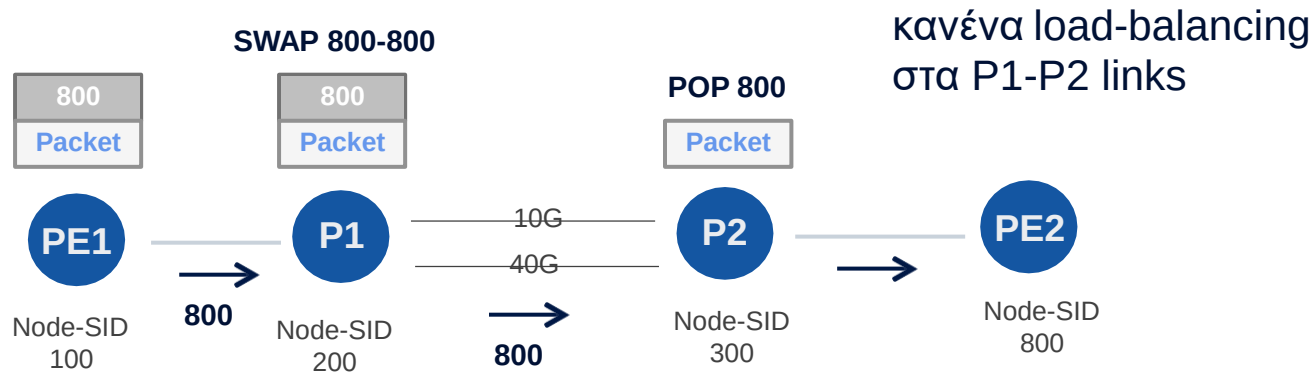
Segment Routing in the Data Center

- BGP-LU ενορχήστρωση των SIDs/labels στο δίκτυο (T- MPLS)
- Controller φτιάχνει paths
- Controller pushes paths
- Controller προγραμματίζει τον egress server



Περίπτωση 4: Adjacency segment load-balancing (1)

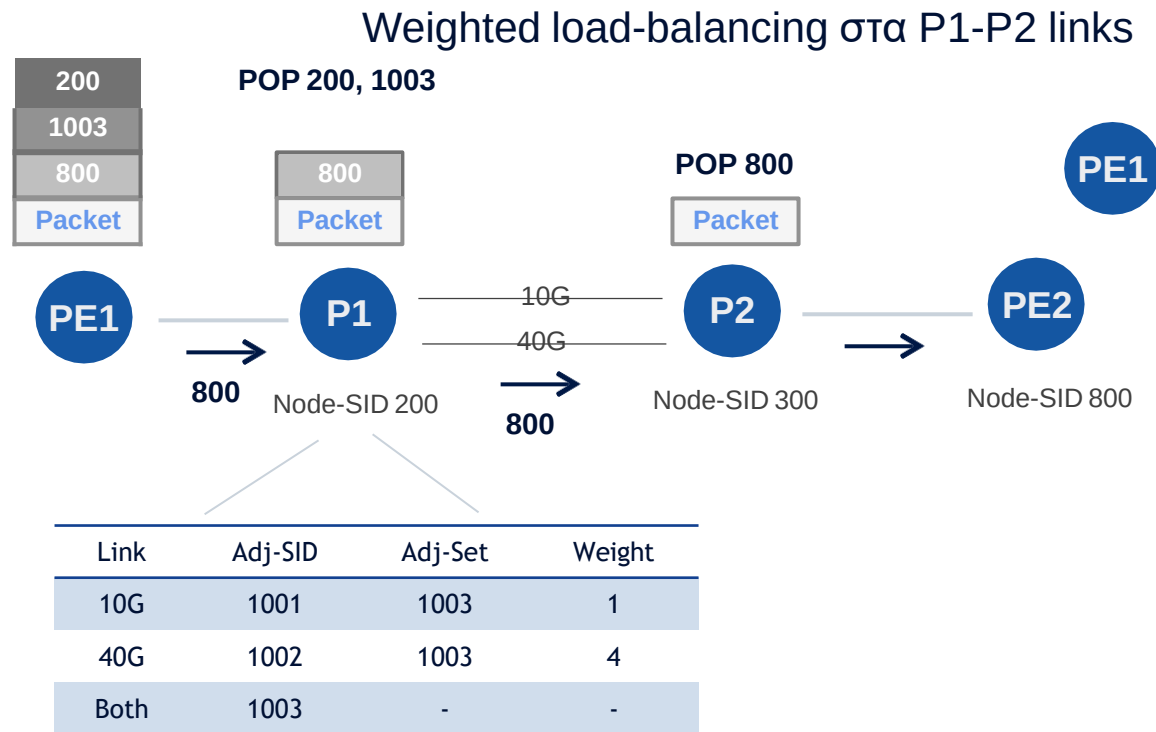
- Δύο adjacencies μεταξύ P1-P2.
- Με την υπόθεση χρήσης capacity-based metrics, το 10G link μεταξύ P1 και P2 είναι αχρησιμοποίητο για shortest path forwarding.



Use case 5: Adjacency segment load-balancing (2)

- Adj-SID TLV προσφέρει δυνατότητα για load-balance μεταξύ πολλαπλών adjacencies.

- P1 ανακοινώνει μεμονωμένα Adj-SIDs για το 10G link (1001) με weight 1, και για το 40G link (1002) με weight 4.
- P1 επίσης ανακοινώνει Adj-SID για το adjacency set (1003)
- PE1 κάνει push segment list {200, 1003, 800}. Node-SID 200 στέλνει κιν/ση -> P1, ενώ με το Adj-SID 1003 γίνεται load-balance κίνησης προς το P2 με αναλογία 4:1.

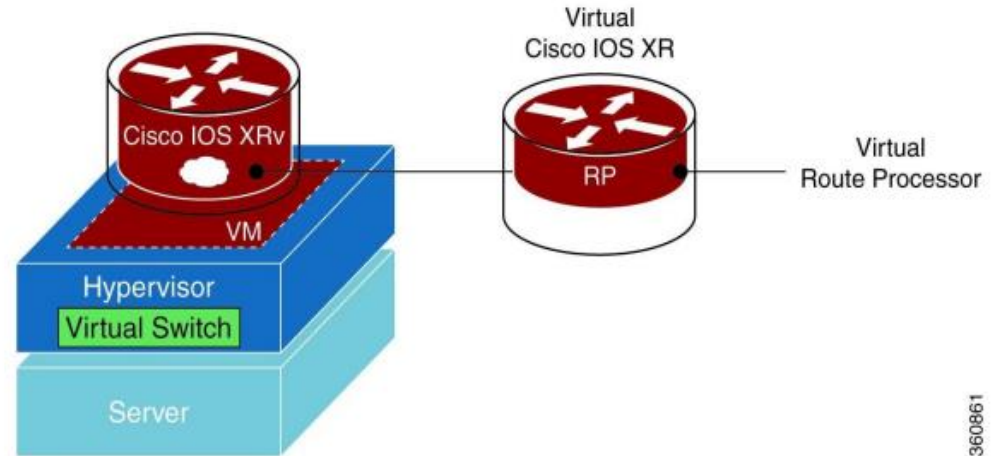




Emulation

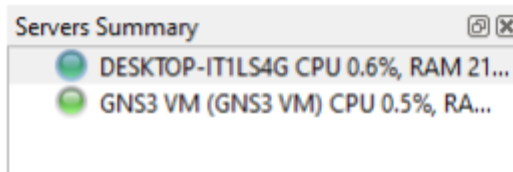
Virtual Box (1)

- Windows 10 PRO vt-d , AMD-V
- Virtualization SW
- GNS3 Local Server
- GNS-GUI VM
- GNS –VM με KVM
- Cisco IOS-XRV



Virtual Box (2) GNS3 (1)

- Nested Virtualization
- GNS-GUI
- GNS3 VM + KVM



GNS3 Local server process

General
Name: GNS3 VM
Operating System: Ubuntu (64-bit)

System
Base Memory: 2048 MB
Processors: 2
Boot Order: Hard Disk, Optical
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Display
Video Memory: 8 MB
Graphics Controller: VBoxVGA
Remote Desktop Server Port: 5966
Recording: Disabled

Storage
Controller: IDE Controller
IDE Primary Device 0: GNS3 VM-disk001.vdi (Normal, 19,53 GB)
Controller: SATA Controller
SATA Port 2: GNS3 VM-disk002.vdi (Normal, 488,28 GB)

Audio
Host Driver: Windows DirectSound
Controller: ICH AC97

Network
Adapter 1: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')
Adapter 2: Intel PRO/1000 MT Desktop (NAT)

USB
Disabled

Shared folders
None

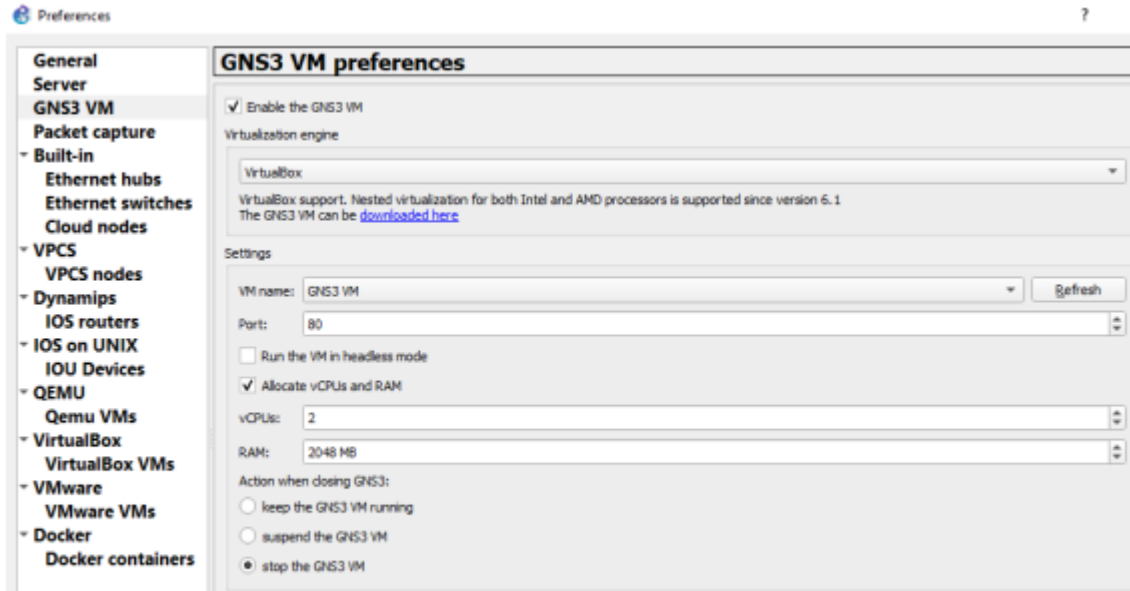
Description
None



GNS3 VirtualBox VM specifications.

Virtual Box (3) GNS3 (3)

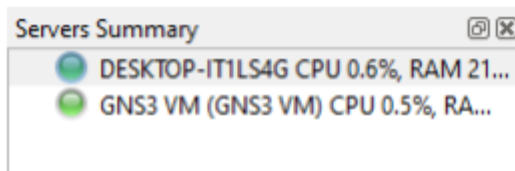
- GNS3 VM



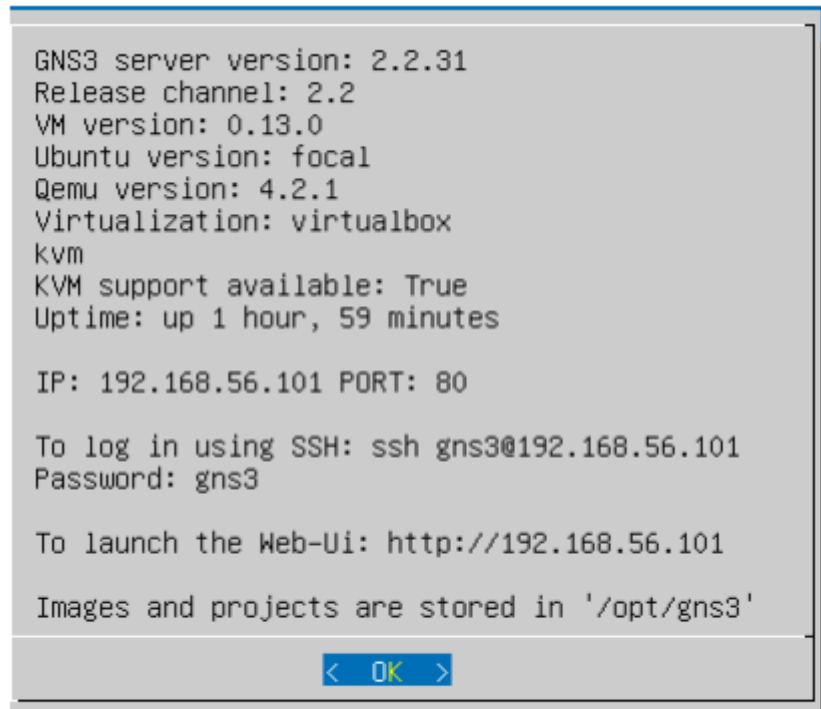
GNS3 VM specifications in GNS3 GUI.

Virtual Box (3) GNS3 (2)

- Ενεργοποίηση KVM στο VM



GNS3 Local server process

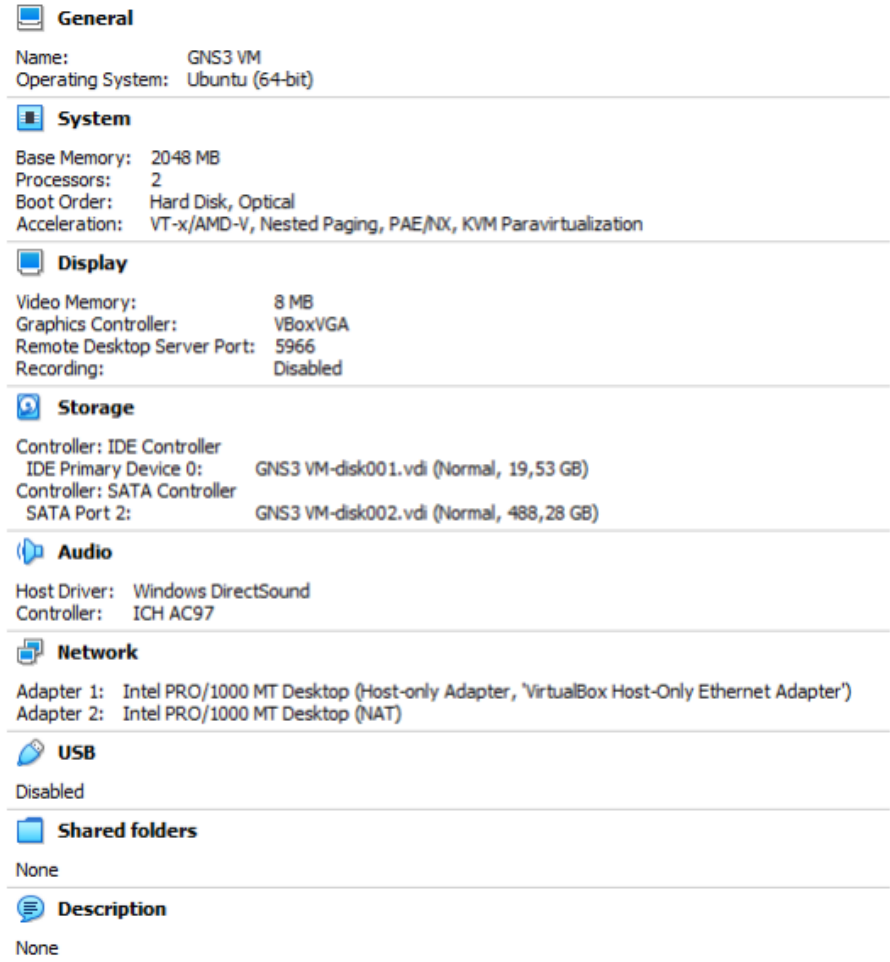


GNS3 VM running in VirtualBox with KVM enabled.

Virtual Box (4) GNS3 (4)

- Create Empty Local template

1. Edit > Preferences > VirtualBox VMs > New.
2. Run this VirtualBox VM on my local computer.
3. From VM list ...recently created CRS1000V
“Use as a link base VM (experimental) option.
4. Finish > Apply > Ok.
5. Edit > Preferences > VirtualBox VMs > CSR1000V > Edit.
6. Console type > telnet.
7. Ok > Apply > Ok
8. Boot
9. Activate Licence



The screenshot displays the Virtual Machine settings window for a VM named 'GNS3 VM'. The settings are organized into several tabs, each with a corresponding icon: General (computer), System (circuit board), Display (monitor), Storage (floppy disk), Audio (speaker), Network (network card), USB (USB plug), Shared folders (folder), and Description (speech bubble). The General tab is selected, showing the VM's name and operating system. The System tab shows memory and processor settings. The Display tab shows video memory and graphics controller. The Storage tab shows two IDE controllers with their respective disk files. The Audio tab shows the host driver and controller. The Network tab shows two network adapters. The USB tab shows the USB controller is disabled. The Shared folders tab shows no shared folders. The Description tab shows no description.

General	
Name:	GNS3 VM
Operating System:	Ubuntu (64-bit)

System	
Base Memory:	2048 MB
Processors:	2
Boot Order:	Hard Disk, Optical
Acceleration:	VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Display	
Video Memory:	8 MB
Graphics Controller:	VBoxVGA
Remote Desktop Server Port:	5966
Recording:	Disabled

Storage	
Controller:	IDE Controller
IDE Primary Device 0:	GNS3 VM-disk001.vdi (Normal, 19,53 GB)
Controller:	SATA Controller
SATA Port 2:	GNS3 VM-disk002.vdi (Normal, 488,28 GB)

Audio	
Host Driver:	Windows DirectSound
Controller:	ICH AC97

Network	
Adapter 1:	Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')
Adapter 2:	Intel PRO/1000 MT Desktop (NAT)

USB	
Disabled	

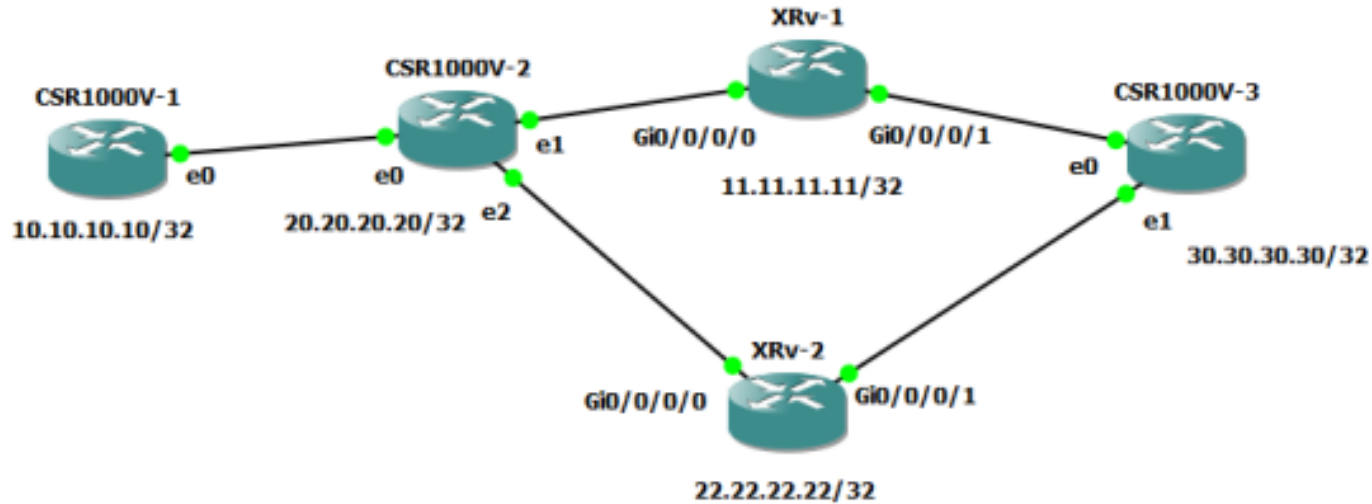
Shared folders	
None	

Description	
None	

GNS3 VirtualBox VM specifications.

Virtual Box - Topology

- CSR-IOS-XR



Virtual Box (4) XRv-1 Basic SR configuration

```
hostname XRv-1
interface Loopback1
  ipv4 address 11.11.11.11 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
  shutdown
!
interface GigabitEthernet0/0/0/0
  ipv4 address 100.100.100.6 255.255.255.252
  shutdown
!
interface GigabitEthernet0/0/0/1
  ipv4 address 100.100.100.13 255.255.255.252
  shutdown
!
interface GigabitEthernet0/0/0/2
  shutdown
!
interface GigabitEthernet0/0/0/3
  shutdown
!
interface GigabitEthernet0/0/0/4
  shutdown
!
interface GigabitEthernet0/0/0/5
  shutdown
!
interface GigabitEthernet0/0/0/6
  shutdown
!
interface GigabitEthernet0/0/0/7
  shutdown
```

```
router isis lab
  is-type level-1
  net 49.0000.0000.0000.0003.00
  address-family ipv4 unicast
  metric-style wide
  segment-routing mpls
!
interface Loopback1
  address-family ipv4 unicast
  prefix-sid index 3
!
!
interface GigabitEthernet0/0/0/0
  circuit-type level-1
  point-to-point
  address-family ipv4 unicast
!
!
interface GigabitEthernet0/0/0/1
  circuit-type level-1
  point-to-point
  address-family ipv4 unicast
!
!
end
```


CSR1000v Basic SR configuration

```
interface Loopback1
ip address 10.10.10.10 255.255.255.255
ip router isis LAB
```

!

```
interface GigabitEthernet0
ip address 100.100.100.1 255.255.255.252
ip router isis LAB
negotiation auto
no mop enabled
no mop sysid
isis circuit-type level-1
isis network point-to-point
```

!

```
segment-routing mpls
```

!

```
connected-prefix-sid-map
address-family ipv4
10.10.10.10/32 index 1 range 1
exit-address-family
```

!

!

```
router isis LAB
net 49.0000.0000.0000.0001.00
is-type level-1
metric-style wide
segment-routing mpls
```

!

- Core facing
- Gig-E ->point-to-point for
- IS-IS IGP



TE configuration CSR1000V-1 and CSR1000V-3 / Head-end

!

mpls traffic-eng tunnels

!

interface GigabitEthernet1

ip address 100.100.100.1 255.255.255.252

ip router isis LAB

mpls traffic-eng tunnels

isis circuit-type level-1

isis network point-to-point

!

router isis LAB

net 49.0000.0000.0000.0001.00

is-type level-1

metric-style wide

segment-routing mpls

mpls traffic-eng router-id Loopback1

mpls traffic-eng level-1

!

segment-routing mpls

!

connected-prefix-sid-map

address-family ipv4

10.10.10.10/32 index 1 range 1

exit-address-family

!

interface Tunnel1

ip unnumbered Loopback1

tunnel mode mpls traffic-eng

tunnel destination 30.30.30.30

tunnel mpls traffic-eng autoroute announce

tunnel mpls traffic-eng priority 6 6

tunnel mpls traffic-eng path-option 10 explicit name SR-PATH segment-routing

!

ip explicit-path name SR-PATH enable

index 1 next-label 16002

index 2 next-label 16004

index 3 next-label 16005

!



TE configuration CSR1000V-1 and CSR1000V-3 / Tail-end

```
!  
mpls traffic-eng tunnels  
!  
interface GigabitEthernet1  
ip address 100.100.100.14 255.255.255.252  
ip router isis LAB  
mpls traffic-eng tunnels  
isis circuit-type level-1  
isis network point-to-point  
!  
!  
interface GigabitEthernet2  
ip address 100.100.100.18 255.255.255.252  
ip router isis LAB  
mpls traffic-eng tunnels  
isis circuit-type level-1
```

```
isis network point-to-point  
!  
router isis LAB  
net 49.0000.0000.0000.0005.00  
is-type level-1  
metric-style wide  
segment-routing mpls  
mpls traffic-eng router-id Loopback1  
mpls traffic-eng level-1  
!  
segment-routing mpls  
!  
connected-prefix-sid-map  
address-family ipv4  
30.30.30.30/32 index 5 range 1  
exit-address-family  
!  
!  
interface Tunnel1  
ip unnumbered Loopback1  
tunnel mode mpls traffic-eng  
tunnel destination 10.10.10.10  
tunnel mpls traffic-eng autoroute announce  
tunnel mpls traffic-eng priority 6 6  
tunnel mpls traffic-eng path-option 10 explicit name SR-PATH segment-routing  
!  
ip explicit-path name SR-PATH enable  
index 1 next-label 16004  
index 2 next-label 16002  
index 3 next-label 16001  
!
```

TE configuration n CSR1000V-1 and XRv2 / Head-Left, Tail Right



```
!
interface Tunnel2
ip unnumbered Loopback1
tunnel mode mpls traffic-eng
tunnel destination 22.22.22.22
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 6 6
tunnel mpls traffic-eng path-option 20 explicit name SR-PATH-XRv2 segment-
routing
!
ip explicit-path name SR-PATH-R4 enable
index 1 next-label 16002
index 2 next-label 16004
!

!
interface Tunnel2
ip unnumbered Loopback1
tunnel mode mpls traffic-eng
tunnel destination 22.22.22.22
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 6 6
tunnel mpls traffic-eng path-option 20 explicit name SR-PATH-XRv2 segment-
routing
!
ip explicit-path name SR-PATH-R4 enable
index 1 next-label 16002
index 2 next-label 16004
!

!
explicit segment-list SR-PATH-CSR1000V1
!
!
```



TE configuration CSR1000V-1 and CSR1000V-3 / Tail-end

```
!  
mpls traffic-eng tunnels  
!  
interface GigabitEthernet1  
ip address 100.100.100.14 255.255.255.252  
ip router isis LAB  
mpls traffic-eng tunnels  
isis circuit-type level-1  
isis network point-to-point  
!  
!  
interface GigabitEthernet2  
ip address 100.100.100.18 255.255.255.252  
ip router isis LAB  
mpls traffic-eng tunnels  
isis circuit-type level-1
```

```
isis network point-to-point  
!  
router isis LAB  
net 49.0000.0000.0000.0005.00  
is-type level-1  
metric-style wide  
segment-routing mpls  
mpls traffic-eng router-id Loopback1  
mpls traffic-eng level-1  
!  
segment-routing mpls  
!  
connected-prefix-sid-map  
address-family ipv4  
30.30.30.30/32 index 5 range 1  
exit-address-family  
!  
!  
interface Tunnel1  
ip unnumbered Loopback1  
tunnel mode mpls traffic-eng  
tunnel destination 10.10.10.10  
tunnel mpls traffic-eng autoroute announce  
tunnel mpls traffic-eng priority 6 6  
tunnel mpls traffic-eng path-option 10 explicit name SR-PATH segment-routing  
!  
ip explicit-path name SR-PATH enable  
index 1 next-label 16004  
index 2 next-label 16002  
index 3 next-label 16001  
!
```

```
ip explicit-path name SR-PATH enable  
index 1 next-label 16004  
index 2 next-label 16002  
index 3 next-label 16001  
!
```

Dynamic TE configuration



```
RP/0/0/CPU0:XRv-2#sh run router isis
router isis LAB
is-type level-1
net 49.0000.0000.0000.0004.00
distribute link-state
!
```

```
RP/0/0/CPU0:XRv-2#sh run segment-routing
!
segment-routing
traffic-eng
!
policy POL1
color 20 end-point ipv4 10.10.10.10
candidate-paths
!
preference 101
dynamic
metric
type te
!
!
!
!
!
!
```



Weight ECMP CSR1000V-1 και XRv-2 με αναλογία 1/3 προς 2/3

```
RP/0/0/CPU0:XRv-2#sh run router isis
router isis LAB
is-type level-1
net 49.0000.0000.0000.0004.00
distribute link-state
address-family ipv4 unicast
metric-style wide
mpls traffic-eng level-1
mpls traffic-eng router-id Loopback1
segment-routing mpls
!
interface Loopback1
address-family ipv4 unicast
prefix-sid index 4
!
!
interface GigabitEthernet0/0/0/0
circuit-type level-1
point-to-point
!
!
interface GigabitEthernet0/0/0/1
circuit-type level-1
point-to-point
!
!
```

"CSR1000V-1(config-srte-policy-path-pref)#explicit segment-list SR-LIST weight

2"

```
RP/0/0/CPU0:XRv-2#sh run segment-routing
segment-routing
traffic-eng
segment-list SR-PATH-1-R1
index 1 mpls label 16002
index 2 mpls label 16001
!
segment-list SR-PATH-2-R1
index 1 mpls label 16005
index 2 mpls label 16003
index 3 mpls label 16002
index 4 mpls label 16001
!
policy POL1
color 20 end-point ipv4 10.10.10.10
candidate-paths
preference 100
explicit segment-list SR-PATH-1-R1
weight 4
!
explicit segment-list SR-PATH-2-R1
weight 2
!
```

SRv6 στο Linux



Kernel 4.10

- Υποστήριξη SRv6 στο Linux kernel από 2017.
- Ενσωμάτωση End, T.Insert6, and T.Encaps6
- Υποστήριξη SRv6 ανά interface. Όλες οι δ/νσεις IPv6 σε ένα SRv6-enabled interface είναι τοπικά local SID
- Iproute2 επέκταση με SR policy.

```
sysctl -w net.ipv6.conf.all.seg6_enabled=1    sysctl -w  
net.ipv6.conf.<device>.seg6_enabled=1  
ip -6 route add <prefix> encap seg6 mode <encapmode> segs <segments> dev <device>
```



Kernel 4.14

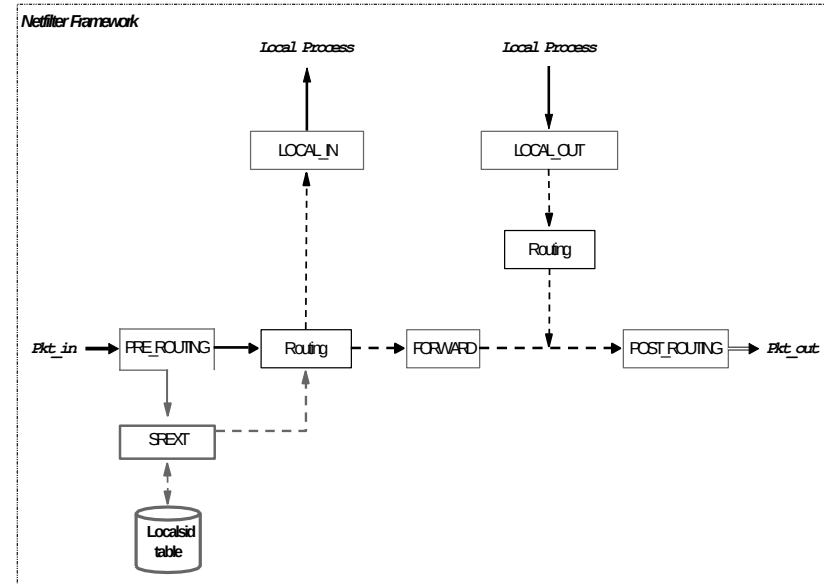
- Περισσότερα SRv6 behaviors :
- `T.Encaps.L2, T.Encaps4, End.X, T.Insert4,`
`End.DX6, End.T, End.DX2, End.DX4,`
`End.DT6, End.B6, End.B6.Encaps.`
- Προσθήκες IProute2

```
ip -6 route add <segment> encap seg6local action  
<action> <params> \ dev <device> table localsid
```

SREXT



- Εξωτερικό kernel module με πρόσθετες λειτουργίες Segment Routing.
- proxy που επιτρέπει σε SR-unaware service/ functions να ενσωματωθούν.



SERA



- SEgment Routing Aware firewall
- SRv6-aware Network function
- με δυνατότητες matching και actions.
- Στο πακέτο, SRH,

Ερωτήσεις