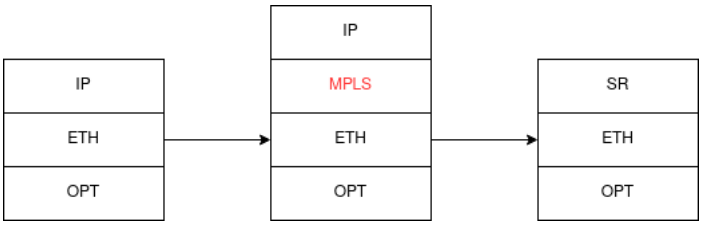
Segment Routing

Fare troubleshooting in MPLS è veramente complicato, perché se guardo un pacchetto l'informazione contenuta nel header trovo una label la quale è un numero, e visto cosi non mi dice nulla riguardo la storia del pacchetto Per ricotruire la storia del pacchetto dovrei andare a ritroso in ogni nodo.

Per questo motivo, MPLS è stato sostituito da Segment Routing nello stack dei protocolli.



l'obiettivo di Segment Routing è quello di tenere tutti i pro di MPLS ed eliminare i contro.

1 Introduction

Segment routing support source routing paradigm. Si assegna un "identifier" chiamati **SID** ,che per ora è un numero, ad ogni nodo della rete e ad ogni entità della rete.

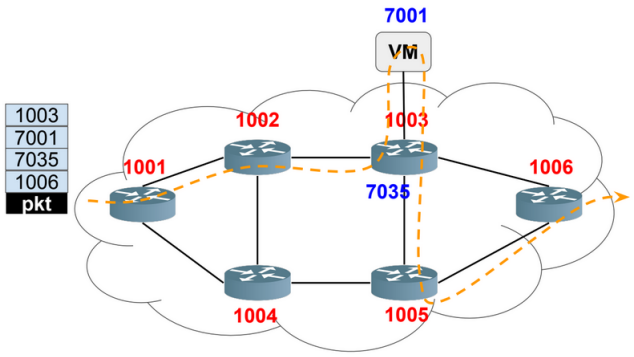
Ogni entità della rete può eseguire due tioologie di istruzioni:

- Topological istructions
- Service based instructions

Le istruzioni possono essere:

- Local: può essere eseguita solo da un entità in modo locale. Un esempio sarebbe quello di inviare un pacchetto su una specifica interfaccia, solo il router proprietario dell'interfaccia può inviare pacchetti su quell'interfaccia
- Global: possono essere eseguite da tutti. Un esempio consiste nell'inviare un pacchetto ad un'altra entita della rete, può essere fatto da tutti

Segment Routing, a differenza di IP permette di creare un percorso nella rete ben specifico. Permette di creare una lista ordinata di destinatari a cui far arrivare il pacchetto. Chiamata **Segment list**



Se il percorso verso una destinazione non è specificato manualmente attraverso la lista viene usato the "shortest path" perché l'unico che conosciamo. Questo perché Segment Routing è una overlay architecture costruita su un underlay infrastructure che in questo caso è IP. Il routing dei pacchetti e gestito dal IP control plane che utilizza come metodologia *the shortest path.*

Il procedimento è il seguente:

1. Segment Routing crea la **segment list**
2. Chiede al control plane quale percorso deve seguire il segmento della segment list
3. Il segmento viene inviato alla destinazione chiesta dal Segment Routing
4. il procedimento si ripete con il segmento successivo

Inserire immagine minuto 27

Come possiamo distribuire le informazioni riguardanti gli altri SIDs della rete? In altri termini, come faccio a sapere dell'esistenza degli altri SIDs della rete?

Questa procedura viene fatta attraverso L'OSPF protocol, quest'ultimo invia un messaggio (LSA, Link state advertisment) agli altri nodi della rete attraverso i suoi link contentente le seguenti info:

- il SID del sender
- I link della rete a cui è collegato
- I SIDs che conosce

Ogni nodo della rete, ha un **topology database** il quale viene riempito con le info in arrivo dagli LSA degli altri SIDs. Una volta che gli altri nodi ricevono LSA di un nodo, lo inoltrano verso gli altri link, come in figura.

Inserire figura al minuto 36

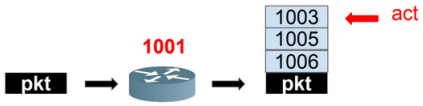
Questo procedimento è chiamato **Flooding**

Il compito di calcolare la segment list è lasciato a un "centralized node" chiamato **SDN controller**. SDN suggerisce di centralizzare l'intelligenza di rete in un componente separato, disassociando il processo di forwarding dei pacchetti (Data Plane) da quello di routing (Control Plane). Calcola i percorsi che devono seguire i pacchetti in rete e installa delle SR policy all'interno dei nodi, le quali decidono dove instradare i pacchetti.

figura al minuti 41

Attualmente Segment Routing non ha sostituito del tutto MPLS e IPv6, per far si che funzioni tutto, SR ingloba le funzioni di entrambi.

1.1 SR forwarding



Quando arriva in ingresso un pacchetto, i nodi SR hanno una forawrding table che permette di capire cosa fare con il pacchetto. Ogni nodo può fare 3 operazioni

- PUSH: Inserisce i segmenti all'inizio della lista dei segmenti
- NEXT: Quando il pacchetto arriva al SID presente nella segment list, il nodo analizza il segmento successivo per capire dove inoltrare il pacchetto senza però eliminare il segmento precedente.
- CONTINUE: Questa operazione è svolta dai nodi di "passaggio". Quelli che non sono presenti nella segment list e ricevono il pacchetto. Consiste in una operazione di inoltro senza modifica della segment list.

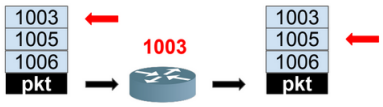


Figure 1: NEXT operation

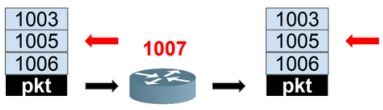


Figure 2: CONTINUE operation

2 Segment Routing Policy

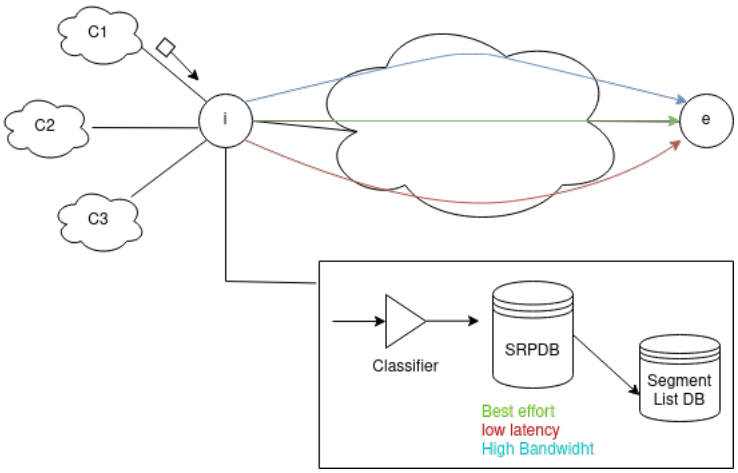
Sono istruzioni logiche che devono gli ingress node devono eseguire nei confronti di un pacchetto in arrivo. Sono implementate negli *(*edge nodes)

Gli edge nodes hanno un database chiamato **SR policy database (SRPDB)** che contiene tutte le policy che devono essere applicate ai rispettivi pacchetti in arrivo.

Ogni *(*tupla) del database ha una struttura di questo tipo:

$$< i, e, color >$$

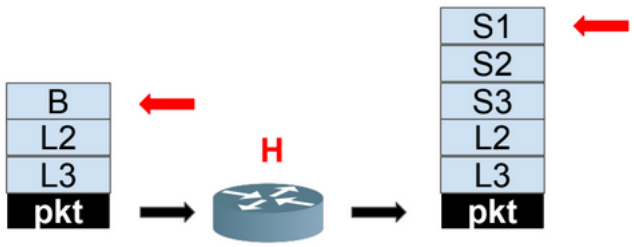
Dove i sta per ingress, e per egress e *color* la tipologia di policy da applicare al pacchetto.



Quando un pacchetto arriva al nodo i avviene questo:

1. Il classifier decide quale policy è meglio per il pacchetto
2. l'SRPDB chiede al Segment List Database (SLDB) quale path tra Quelli del colore scelto dal classifier sia meglio per il pacchetto
3. SRPDB scrive la tupla nel suo database

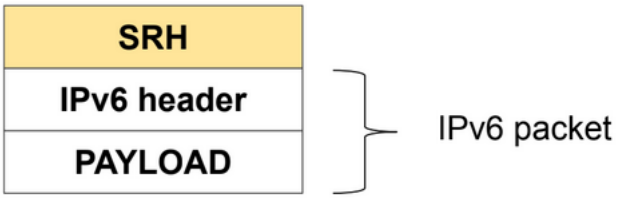
2.1 Bindig SID



Ogni tupla del SRPDB può essere assegnata ad uno specifico SID. Quando si riceve in ingresso una segment list, dove all'interno di essa di essa c'è un SID **(B)**, il quale è assegnato ad una policy del database, Il segmento corrispondente, viene sostituito con la segment list associata alla policy di **B**. Il *bindng SID* è utile perché permette di ridurre la dimensione della segment list.

3 SRv6 and Segment Routing extension header

Come detto prima, Segment Routing è implementato in overlay ad MPLS e IPv6, per favorire la transizione.



Per implementare SR in IPv6, viene aggiunto un header SRH (Segment Routing Header) gestito come se fosse un extension header di IPv6.

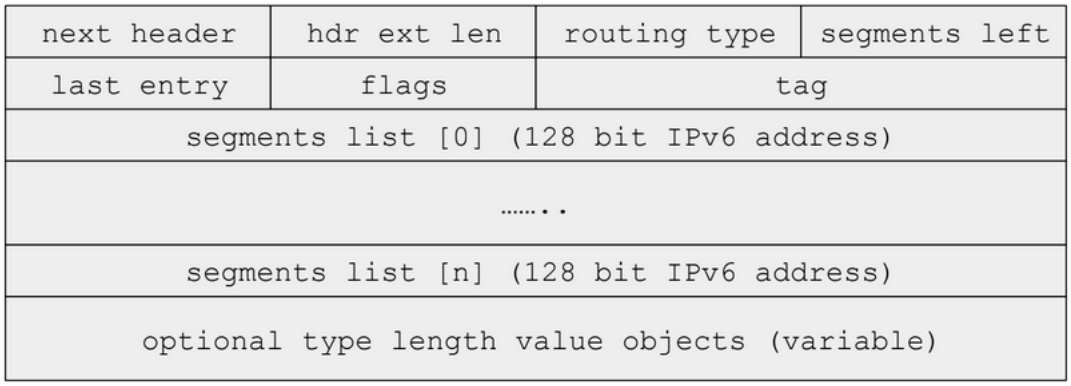


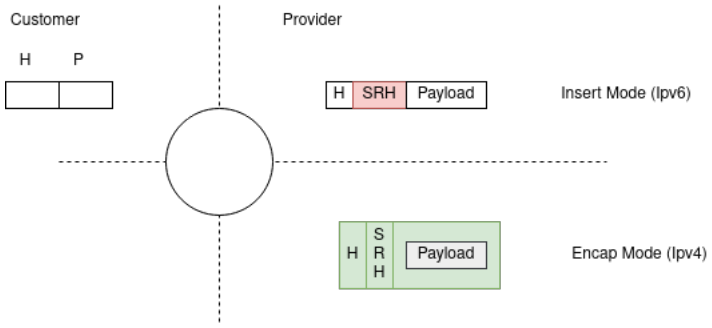
Figure 3: SRv6 Extension Header

Il campo ¡seg left¿ è un puntatore allo stack dei segmenti.

L'extension header di SG tratta i SID come indirizzi IPv6 incrementando di gran lunga la dimensione del header. In MPLS invece, le label hanno di dimensione 20 bit.

Ci sono due modi per aggiungere un extension header ad un pacchetto proveniente da un costumer. Dipende da quale protocollo IP viene usato dal costumer

- Se IPv6: l'extension header viene inserito in coda e trattato come se fosse un extension header normale.
- Se IPv4: il pacchetto viene incapsulato



3.1 Local Sid Table

Ogni nodo Srv6 mantiene localmente una tabella locale dei SID, la quale per ognuno di essi, è associata una funzione, chiamata **end behavior**

MyLocalSID Table

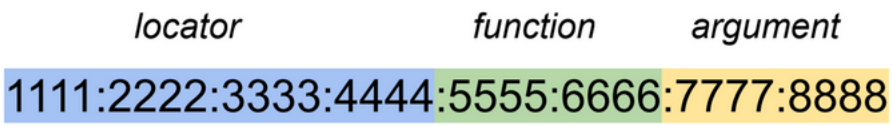
SID	instruction
...	...

Le funzioni possono essere di due tipologie:

- **End**: È una funzione associata quando il SID di destinazione si trova nella MyLocalSID Table
- **End.X** è una funzione di Layer 3

4 Network Programming

Grazie alle caratteristice spiegante precedentemente è possibile creare una programmazione di rete, per cui ad ogni SID è possibile associare una funzione corrispondente.



In figura è rappresentata la struttura di un indirizzo IPv6 utilizzato in particolare nel protocollo, SRv6. I primi 64 bits dell'inidirizzo vengono uasti come localizzatore del SID. I campi *function* e *argument* sono campi usati dalla MyLocalSID Table del *locator* per inidirizzare il pacchetto verso il nodo che performerà la funzione corrispondente.