

Configurazione di una Rete con VLAN e Routing Inter-VLAN

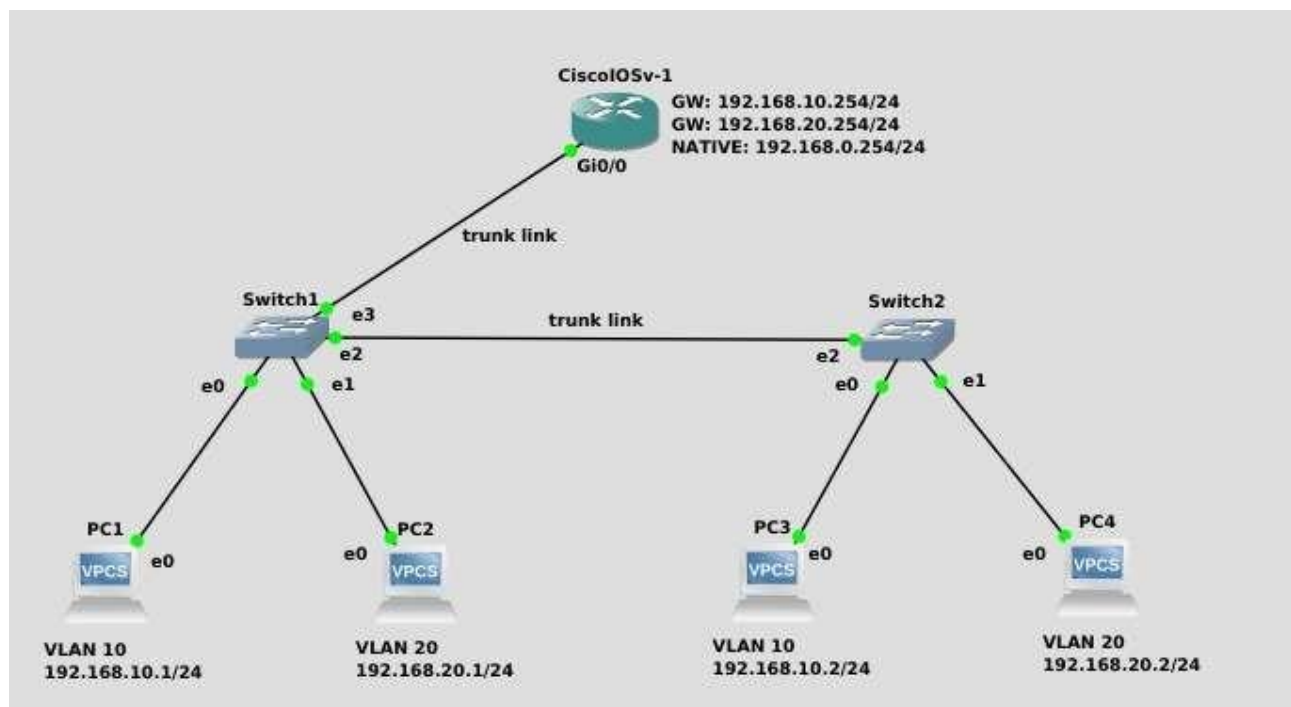
Relazione dell'Elaborato per il corso di Reti di Telecomunicazione

Autori: Albertini Giulia, Cinelli Lorenzo, Osti Kimi

INTRODUZIONE

Progettare e configurare una topologia virtuale di rete su GNS3, che includa due VLAN separate e ne implementi la comunicazione. Gli obiettivi principali sono la realizzazione del routing inter-VLAN, la verifica del suo funzionamento e l'analisi dei pacchetti osservati.

TOPOLOGIA



Nella topologia realizzata, sono presenti quattro terminali, collegati a un solo Router tramite Switch, ma suddivisi in due network IP. L'obiettivo della configurazione è quello di mappare le due network su due VLAN separate, limitando i domini di broadcast senza però perdere connettività fra i terminali, che devono comunque essere tutti in grado di comunicare. Nel farlo, si mappano le porte di entrambi gli Switch al fine di supportare entrambe le VLAN (di conseguenza richiedendo un adeguato collegamento fra Switch per permettere la comunicazione fra terminali sulla stessa VLAN) e si configura il router in modo da mappare sulla stessa interfaccia fisica tutte le interfacce legate alle VLAN del sistema.

CONFIGURAZIONE

1. Switch

Nella configurazione degli switch, al fine di supportare VLAN, si distinguono due tipi di porte: **Access** rivolte agli endpoint in grado di trasportare i pacchetti di una sola VLAN, e **Trunk** rivolte ai dispositivi di instradamento (router o altri switch) in grado di trasportare i pacchetti di tutte le VLAN, identificati da appositi tag.

Switch 1

Port	VLAN	Type	EtherType
0	10	access	
1	20	access	
2	1	dot1q	
3	1	dot1q	

Nello Switch 1, oltre alle porte Access ripartite fra le due VLAN, si predispongono due porte Trunk. A queste verranno collegati un router e uno Switch.

Switch 2

Port	VLAN	Type	EtherType
0	10	access	
1	20	access	
2	1	dot1q	

Nello Switch 2, invece, è predisposta un'unica porta Trunk dedicata al collegamento con lo Switch 1.

In particolare, gli Switch utilizzati per il progetto supportano porte Trunk basate sul protocollo IEEE 802.1Q (abbreviato dot1q), basato sull'associazione di specifici tag ai pacchetti appartenenti a VLAN diverse, in modo da separare i domini di broadcast pur condividendo il canale. In assenza dei tag, non vi sarebbe infatti modo per uno Switch di sapere da quale porta, e pertanto da quale VLAN, provenga il traffico che gli viene trasmesso dall'altro Switch.

2. PC

I PC sono divisi in due network (192.168.10.0/24 e 192.168.20.0/24). Per separare in maniera più efficace i due domini, e in particolare per mappare le due network su domini di broadcast differenti, gli Switch sono già stati predisposti per accettare collegamenti su porte Access sufficienti ad ospitare i PC della topologia su due VLAN separate.

In fase preliminare, in assenza di un gateway, si associano solamente gli indirizzi ai PC delle due network, con il comando `ip <ip-addr>/<netmask>` ripetuto su ogni host.

```
PC1> sh ip
NAME       : PC1[1]
IP/MASK    : 192.168.10.1/24
GATEWAY    : 0.0.0.0
DNS        :
MAC        : 00:50:79:66:68:03
LPORT      : 10008
RHOST:PORT : 127.0.0.1:10009
MTU        : 1500
```

Testing

Mostrando l'interfaccia Ethernet di uno dei PC si vede come abbia accettato l'indirizzo IP, ma non abbia ancora un gateway predefinito. Si procede a testare la connettività Intra-VLAN (e quindi il corretto funzionamento del collegamento Trunk tra i due Switch) con il comando `ping 192.168.10.2`.

```
PC1> ping 192.168.10.2
84 bytes from 192.168.10.2 icmp_seq=1 ttl=64 time=1.438 ms
84 bytes from 192.168.10.2 icmp_seq=2 ttl=64 time=1.074 ms
84 bytes from 192.168.10.2 icmp_seq=3 ttl=64 time=1.401 ms
84 bytes from 192.168.10.2 icmp_seq=4 ttl=64 time=1.032 ms
84 bytes from 192.168.10.2 icmp_seq=5 ttl=64 time=1.156 ms
```

Per provare che i due domini di broadcast siano effettivamente separati, si simula un tentativo di IP spoofing: si assegna a un PC della VLAN 10 un indirizzo della network 192.168.20.0 e si tenta un ping verso un pc della VLAN 20, associata a tale network.

```
PC1> ip 192.168.20.3/24
Checking for duplicate address...
PC1 : 192.168.20.3 255.255.255.0

PC1> sh ip
NAME       : PC1[1]
IP/MASK    : 192.168.20.3/24
GATEWAY    : 0.0.0.0
DNS        :
MAC        : 00:50:79:66:68:03
LPORT     : 10008
RHOST:PORT : 127.0.0.1:10009
MTU        : 1500
```

```
PC1> ping 192.168.20.1
host (192.168.20.1) not reachable
```

Il tentativo di spoofing non va a buon fine, dimostrando che i due domini di broadcast sono stati separati correttamente: PC1, della VLAN10, non riesce infatti a risolvere l'indirizzo IP del PC2 perché la sua ARP request non viene ricevuta dagli host della VLAN20.

3. Router

Lavorando con un router Cisco, lo si può configurare tramite il comando `configure terminal`. Una volta dentro il terminale di configurazione, la prima operazione da fare è l'attivazione dell'interfaccia fisica.

```

Router(config)#int Gi0/0
Router(config-if)#no shut
Router(config-if)#
*Dec  9 17:18:46.376: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
Router(config-if)#
*Dec  9 17:18:47.375: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
Router(config-if)#exit

```

Dopo aver attivato l'interfaccia fisica, si procede configurando le interfacce logiche. In particolare, se ne predispongono tre: una per ciascuna delle due VLAN, e una terza per la VLAN nativa (nel nostro caso la VLAN 1) associata ai canali Trunk.

```

Router(config)#int Gi0/0.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip add 192.168.10.254 255.255.255.0
Router(config-subif)#exit
Router(config)#
Router(config)#
Router(config)#int Gi0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip add 192.168.20.254 255.255.255.0
Router(config-subif)#exit

```

Configurando le interfacce logiche delle VLAN 10 e 20, si usano indirizzi IP delle network a cui appartengono i relativi host. Questo perché il router verrà usato da esse come gateway per la comunicazione inter-VLAN.

```

Router(config)#int Gi0/0.1
Router(config-subif)#encapsulation dot1Q 1 native
Router(config-subif)#ip add 192.168.0.254 255.255.255.0
Router(config-subif)#end
Router#
*Dec  9 17:25:28.755: %SYS-5-CONFIG_I: Configured from console by console

```

Nella configurazione dell'interfaccia logica legata alla VLAN nativa si usa invece un indirizzo IP appartenente a una terza network separata.

Testing

Dopo aver configurato il router, si associano staticamente gli IP delle interfacce logiche appena configurate come default gateway dei PC delle rispettive VLAN. Per testare il corretto funzionamento della configurazione del router, si tenta un ping fra host di VLAN diverse. In particolare, si usa il PC4 per fare ping verso il PC1.

```

PC4> ping 192.168.10.1

192.168.10.1 icmp_seq=1 timeout
84 bytes from 192.168.10.1 icmp_seq=2 ttl=63 time=3.056 ms
84 bytes from 192.168.10.1 icmp_seq=3 ttl=63 time=2.828 ms
84 bytes from 192.168.10.1 icmp_seq=4 ttl=63 time=2.575 ms
84 bytes from 192.168.10.1 icmp_seq=5 ttl=63 time=3.833 ms

```

Osservando il risultato, si evidenzia la perdita del primo pacchetto a causa di un timeout. Si ripete quindi il ping per verificare se la problematica si ripete.

```
PC4> ping 192.168.10.1

84 bytes from 192.168.10.1: icmp_seq=1 ttl=63 time=3.506 ms
84 bytes from 192.168.10.1: icmp_seq=2 ttl=63 time=2.680 ms
84 bytes from 192.168.10.1: icmp_seq=3 ttl=63 time=2.480 ms
84 bytes from 192.168.10.1: icmp_seq=4 ttl=63 time=12.548 ms
84 bytes from 192.168.10.1: icmp_seq=5 ttl=63 time=6.175 ms
```

Si nota che il problema non si ripete, ma che anzi fin da subito i tempi si allineano a quelli degli ultimi pacchetti del ping precedente. Si suppone quindi che il pacchetto sia stato perso nell'effettuare una ARP request, operazione necessaria a PC4 per registrare nella propria ARP table la traduzione da indirizzo IP a indirizzo MAC del suo default gateway.

CATTURA DEL TRAFFICO

Le catture si sono svolte usando Wireshark su macchina virtuale Linux, dove si è installata la versione stand-alone di GNS3. Si analizzano qui le comunicazioni che si sono registrate in fase di avvio dei nodi della topologia, così come in corrispondenza di due ping.

Catturando i pacchetti, si è notato che la comunicazione verso il server GNS3 viene incapsulata interamente dietro a uno scambio di pacchetti sull'interfaccia di Loopback. In particolare, i pacchetti scambiati fra i nodi della configurazione vengono incapsulati in pacchetti UDP e i pacchetti di controllo per le funzioni applicative vengono incapsulati in pacchetti TCP. La loro funzione (e di conseguenza struttura) vengono analizzate più a fondo in seguito.

Avvio dei nodi

All'avvio dei nodi, si registrano principalmente pacchetti TCP, con alcuni pacchetti HTTP all'inizio per la comunicazione di informazioni (tramite JSON) e ICMP che segnalano porte non raggiungibili. Si vedono anche alcuni pacchetti UDP, che verranno analizzati meglio in seguito e rappresentano comunicazioni fra le interfacce dei nodi virtuali.

Per questioni di comprensibilità, avendo visto una ricorrenza nelle sequenze dei pacchetti soprattutto delle prime fasi, si filtrano i pacchetti al fine di analizzare una sessione di comunicazione. Sapendo che il server GNS3 si trova dietro la porta TCP 3080, ci si concentra con la sessione aperta con la porta client 39760.

(tcp.srcport == 3080 && tcp.dstport == 39760) (tcp.srcport == 39760 && tcp.dstport == 3080)						
No.	Time	Source	Destination	Protocol	Length	Info
26	3.665156201	127.0.0.1	127.0.0.1	TCP	74	39760 → 3080 [SYN] Seq=0 Win=0
27	3.665168818	127.0.0.1	127.0.0.1	TCP	74	3080 → 39760 [SYN, ACK] Seq=645114400
28	3.665178179	127.0.0.1	127.0.0.1	TCP	66	39760 → 3080 [ACK] Seq=1 Ack=645114400
35	3.666276795	127.0.0.1	127.0.0.1	HTTP	472	POST /v2/compute/projects/f5k
36	3.666287748	127.0.0.1	127.0.0.1	TCP	66	3080 → 39760 [ACK] Seq=1 Ack=645114400
109	3.927436220	127.0.0.1	127.0.0.1	TCP	297	3080 → 39760 [PSH, ACK] Seq=1 Ack=645114400
110	3.927450569	127.0.0.1	127.0.0.1	TCP	66	39760 → 3080 [ACK] Seq=407 Ack=645114400
111	3.927482260	127.0.0.1	127.0.0.1	HTTP/JSON	520	HTTP/1.1 200 OK, JSON (appli
112	3.927486201	127.0.0.1	127.0.0.1	TCP	66	39760 → 3080 [ACK] Seq=407 Ack=645114400
115	3.928569964	127.0.0.1	127.0.0.1	TCP	66	39760 → 3080 [FIN, ACK] Seq=407 Ack=645114400
116	3.929483356	127.0.0.1	127.0.0.1	TCP	66	3080 → 39760 [FIN, ACK] Seq=645114400
117	3.929493804	127.0.0.1	127.0.0.1	TCP	66	39760 → 3080 [ACK] Seq=408 Ack=645114400

1. Sequenza di Handshake

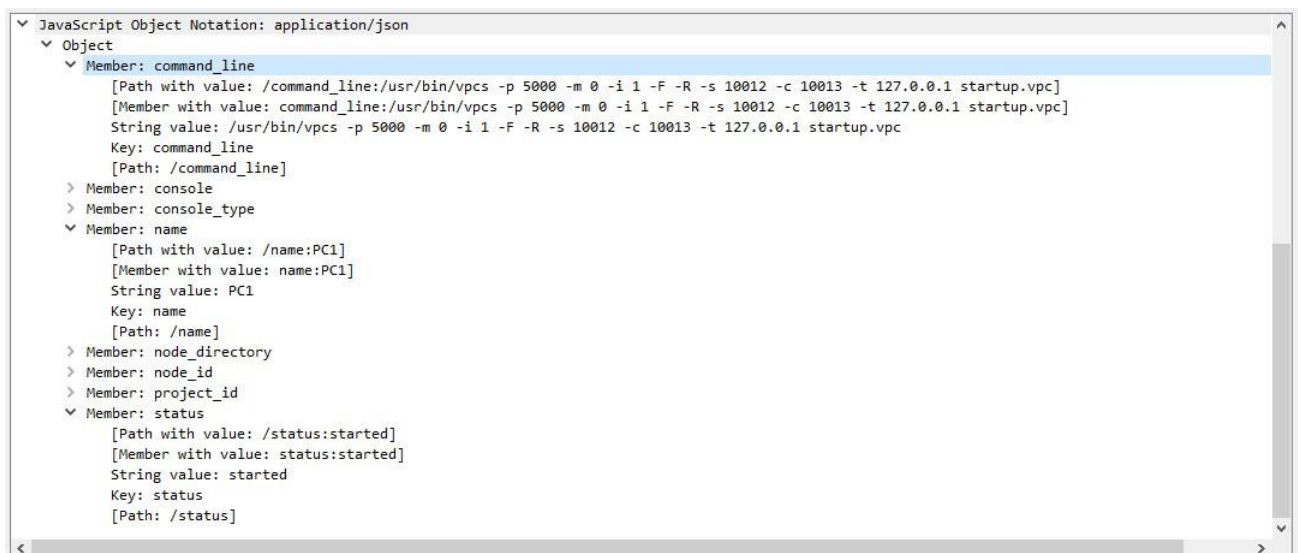
Nei pacchetti dal 26 al 28 si osserva la classica sequenza di handshake delle sessioni TCP. Le due porte qui si scambiano una serie di pacchetti con flag SYN, SYN/ACK e ACK attivi per segnalare la corretta ricezione dei pacchetti e accordarsi sui numeri di sequenza e di acknowledge.

2. HTTP POST

Il pacchetto 35 è una richiesta HTTP POST verso il server GNS3. In particolare, qui vengono inseriti i dati di autenticazione del nodo per identificarsi presso il server, e richiedere di conseguenza l'attivazione della propria configurazione di avvio.

3. HTTP RESPONSE

Il pacchetto 111 contiene la risposta alla precedente richiesta HTTP. In particolare, il messaggio "200 – OK" sottolinea come la richiesta sia andata a buon fine, e Wireshark stesso segnala la presenza di un JSON all'interno della risposta per comunicare i dati del nodo che ha svolto la richiesta.



```
JavaScript Object Notation: application/json
Object
  Member: command_line
    [Path with value: /command_line:/usr/bin/vpcs -p 5000 -m 0 -i 1 -F -R -s 10012 -c 10013 -t 127.0.0.1 startup.vpc]
    [Member with value: command_line:/usr/bin/vpcs -p 5000 -m 0 -i 1 -F -R -s 10012 -c 10013 -t 127.0.0.1 startup.vpc]
    String value: /usr/bin/vpcs -p 5000 -m 0 -i 1 -F -R -s 10012 -c 10013 -t 127.0.0.1 startup.vpc
    Key: command_line
    [Path: /command_line]
  Member: console
  Member: console_type
  Member: name
    [Path with value: /name:PC1]
    [Member with value: name:PC1]
    String value: PC1
    Key: name
    [Path: /name]
  Member: node_directory
  Member: node_id
  Member: project_id
  Member: status
    [Path with value: /status:started]
    [Member with value: status:started]
    String value: started
    Key: status
    [Path: /status]
```

La lettura dei campi chiave del JSON permette di identificare, dietro questa porta TCP, la richiesta di attivazione di PC1, che in questo caso è andata a buon fine come riportato nel campo status, con valore "started".

4. Sequenza di chiusura

Nei pacchetti dal 115 al 117, appaiono i pacchetti standard della sequenza di chiusura della comunicazione TCP. In particolare, si sottolinea la presenza di flag FIN/ACK e ACK per confermare la ricezione del FIN/ACK.

In particolare, si evidenziano in questa fase le porte TCP 39760 (PC1), 39762 (PC2), 51760 (PC3), 39764 (PC4), 51762 (router).

Ping intra-VLAN

201	9.615889916	127.0.0.1	127.0.0.1	UDP	140	10012 → 10013	Len=98
202	9.615939061	127.0.0.1	127.0.0.1	UDP	140	10000 → 10001	Len=98
203	9.616062158	127.0.0.1	127.0.0.1	UDP	144	10008 → 10009	Len=102
204	9.616085030	127.0.0.1	127.0.0.1	UDP	140	10005 → 10004	Len=98
205	9.616119165	127.0.0.1	127.0.0.1	UDP	140	10017 → 10016	Len=98
206	9.616180132	127.0.0.1	127.0.0.1	UDP	140	10016 → 10017	Len=98
207	9.616205541	127.0.0.1	127.0.0.1	UDP	140	10004 → 10005	Len=98
208	9.616235869	127.0.0.1	127.0.0.1	UDP	144	10009 → 10008	Len=102
209	9.616272938	127.0.0.1	127.0.0.1	UDP	140	10001 → 10000	Len=98
210	9.616305966	127.0.0.1	127.0.0.1	UDP	140	10013 → 10012	Len=98

Si analizza qui un tentativo di ping intra-VLAN, in particolare fra il PC1 e il PC3, entrambi della VLAN 10. Dall'analisi dei pacchetti, si evince che ogni pacchetto identifica un passaggio di interfaccia, che sia tra apparati differenti o all'interno del medesimo dispositivo. La comunicazione in ogni direzione si compone di 5 salti, quindi per un ping completo ne servono 10. Si vede infatti che le porte dei primi pacchetti sono ripercorse esattamente al contrario per i secondi 5. Contando i salti e ripercorrendo le interfacce, si nota come il salto fra le interfacce logiche del router non venga tracciato da questi pacchetti. Ciò perché, seppur cambiando interfaccia logica, la trama rimane sempre sulla stessa interfaccia fisica.

Si evidenzia inoltre come nel passaggio tra i due switch, che avviene mediante un collegamento Trunk, il pacchetto sia di dimensione maggiore rispetto agli altri hop. Questo è dato dall'utilizzo di tag per identificare a quale VLAN il pacchetto appartiene.

Si nota poi, come particolare, che il pacchetto di rete inviato dagli host virtuali in GNS3 viene fisicamente incapsulato all'interno di un pacchetto UDP su interfaccia di Loopback. Questo significa che, all'esterno del pacchetto (comprensivo di header) vengono posti degli ulteriori header per ottenere l'effetto di incapsulamento desiderato. Il contenuto del pacchetto originale viene quindi completamente interpretato come campo dati da Wireshark, rendendone parzialmente più complicata la lettura. Si osserva il contenuto del pacchetto UDP, evidenziando la parte dati.

0000	0	0	0	0	0	0	0	0	0	0	0	0	8	0	69	0
0010	0	126	37	247	64	0	64	17	22	118	127	0	0	1	127	0
0020	0	1	39	28	39	29	0	106	254	125	0	80	121	102	104	2
0030	0	80	121	102	104	0	8	0	69	0	0	84	193	216	0	0
0040	64	1	35	125	192	168	10	1	192	168	10	2	8	0	71	73
0050	216	193	0	1	8	9	10	11	12	13	14	15	16	17	18	19
0060	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
0070	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
0080	52	53	54	55	56	57	58	59	60	61	62	63				

Qui si vede intanto come la dimensione della parte dati sia 98byte, contro gli 84 indicati da GNS3 quando si esegue il comando ping. Questo avviene perché qui vengono considerati nella dimensione anche i byte di header Ethernet, mentre nella dimensione indicata da GNS3 questi byte non vengono considerati. Si evidenziano qui alcuni byte salienti:

1. **Byte 003b:** valore 84. Indica la dimensione totale del pacchetto IPv4 associata al pacchetto ICMP echo request (header compreso). Questo è il valore indicato dal comando ping all'interno di GNS3, dove non viene calcolata la lunghezza dell'header Ethernet, pari a 14byte.

2. **Byte da 0044 a 004b:** valori 192.168.10.1 – 192.168.10.2. Sono gli indirizzi IP degli host coinvolti nel ping, indicati all'interno dell'header IP.

Ping inter-VLAN

242	10.650038898	127.0.0.1	127.0.0.1	UDP	140	10012 → 10013	Len=98
243	10.650266839	127.0.0.1	127.0.0.1	UDP	140	10000 → 10001	Len=98
244	10.650291159	127.0.0.1	127.0.0.1	UDP	144	10010 → 10011	Len=102
245	10.650310595	127.0.0.1	127.0.0.1	UDP	144	10021 → 10020	Len=102
246	10.650733007	127.0.0.1	127.0.0.1	UDP	144	10020 → 10021	Len=102
247	10.651385961	127.0.0.1	127.0.0.1	UDP	144	10011 → 10010	Len=102
248	10.651412428	127.0.0.1	127.0.0.1	UDP	144	10008 → 10009	Len=102
249	10.651430553	127.0.0.1	127.0.0.1	UDP	140	10007 → 10006	Len=98
250	10.651448858	127.0.0.1	127.0.0.1	UDP	140	10019 → 10018	Len=98
251	10.651480154	127.0.0.1	127.0.0.1	UDP	140	10018 → 10019	Len=98
252	10.651489403	127.0.0.1	127.0.0.1	UDP	140	10006 → 10007	Len=98
253	10.651499367	127.0.0.1	127.0.0.1	UDP	144	10009 → 10008	Len=102
254	10.651509138	127.0.0.1	127.0.0.1	UDP	144	10010 → 10011	Len=102
255	10.651518690	127.0.0.1	127.0.0.1	UDP	144	10021 → 10020	Len=102
256	10.651825490	127.0.0.1	127.0.0.1	UDP	144	10020 → 10021	Len=102
257	10.651838597	127.0.0.1	127.0.0.1	UDP	144	10011 → 10010	Len=102
258	10.651854391	127.0.0.1	127.0.0.1	UDP	140	10001 → 10000	Len=98
259	10.651868521	127.0.0.1	127.0.0.1	UDP	140	10013 → 10012	Len=98

Si analizza qui un tentativo di ping inter-VLAN, in particolare fra il PC1 della VLAN 10 e il PC4 della VLAN 20. In questo caso la comunicazione in ciascuna direzione si compone di 9 salti, perché si deve passare tramite il router in quanto il pacchetto viaggia tra due differenti VLAN. Avendo identificato dalla cattura precedente il salto 10000 => 10001 come il salto di entrata in Switch1 (dopo di esso il pacchetto è più lungo e pertanto tagged) e il salto 10008 => 10009 come il salto fra le interfacce dei due Switch (giustificato dal fatto che il salto successivo ripristina la lunghezza originale del pacchetto, andando quindi verosimilmente a rimuovere il tag), rimangono da analizzare gli hop 10010 => 10011 e 10020 => 10021 (e gli opposti). Con ogni probabilità, questi hop sono dovuti allo spostamento del pacchetto verso l'interfaccia che si collega al router, alla trasmissione al router, alla sua ricezione e alla traduzione del tag per predisporre la comunicazione verso Switch2 con il tag della nuova VLAN di appartenenza. Dopodiché, il pacchetto esce verso Switch2 con il salto 10008 => 10009 come in precedenza e il tag viene rimosso, riportandosi alla lunghezza originaria.

Si nota come anche gli hop che avvengono all'interno del router siano tagged proprio per identificare la VLAN di appartenenza della trama. Come nel caso precedente le porte dei primi 9 pacchetti vengono percorse precisamente al contrario dai secondi 9.