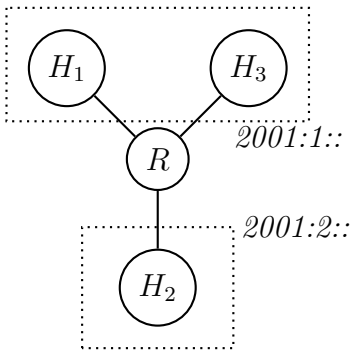
Nome: *Tommaso* Cognome: *Botarelli* Matricola: *7136210*

1 Obiettivo

L'obiettivo di questo report è quello di riassumere quanto prodotto per effettuare una simulazione dell'autoconfigurazione in una piccola rete IPV6 e commentare i risultati della simulazione stessa.

2 Struttura della rete e dell'esperimento

La rete presa in esame è la seguente:



In particolare i due hosts H_1 e H_3 hanno uno stesso network prefix (2001 : 1 ::), mentre H_2 presenta un network prefix diverso (2001 : 2 ::).

Nell'ambito di questa simulazione quello che vogliamo è raggiungere l'autoconfigurazione (configurare un ip address) di tutti gli hosts e osservare un ping all'interno della rete.

In particolare come illustrato in seguito si avranno due ping:

- da H_1 ping di H_3;
- da H_2 ping di H_1;

Questi due ping permettono quindi di testare l'arrivo e l'invio di messaggi sia fra nodi della subnet con stessa network prefix sia fra nodi con network prefix distinte.

3 Simulazione NS3

Per effettuare l'analisi necessaria è stata costruita una simulazione grazie all'utilizzo di *NS3*.

Il codice e i file *.pcap* di output possono essere visionati nel seguente *repo*.

Per poter rieseguire l'esperimento quello che occorre fare è:

1 - configurare NS3 come descritto nel *getting started*;

2 - clonare il contenuto del repo in una nuova cartella all'interno del progetto NS3;

3 - all'interno del file *CMakeLists.txt* di NS3 aggiungere la cartella alla build;

4 - lanciare l'esperimento una volta completata la build;

La rete è stata creata seguendo i seguenti passaggi (vedi *assigment_1*):

- creazione dei nodi (*Node*);
- creazione delle reti contenenti i nodi (*NodeContainer*);
- installazione nella rete complessiva di tutti i nodi e del router dello stack IPV6 (*InternetStackHelper*);
- creazione del canale CSMA e dei device di interfaccia al canale nelle due sotto reti;
- settaggio del prefix delle due reti e specifica del nodo router;
- configurazione per il router advertisement;
- creazione dei due ping. In particolare si configura il ping di H_2 da H_1 con 3 pacchetti nell'intervallo di simulazione $[2.0, 7.0]$ e il ping di H_1 da H_3 con analogo numero di pacchetti nell'intervallo di simulazione $[8.0, 15.0]$;
- lancio della simulazione e scrittura dei risultati in file con estensione *.pcap*;

4 Risultati

Una volta eseguita una simulazione con NS3 è possibile visualizzare i file con estensione *.pcap* con programmi come *Wireshark*.

NS3 genera per ogni dispositivo di rete del nodo un file *.pcap* diverso che permette quindi di visualizzare tutti i pacchetti in ingresso e uscita di quel dispositivo.

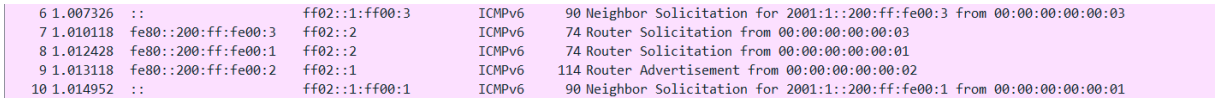
Prendendo come esempio il file *R_subnet_1.pcap* che mostra i pacchetti per l'interfaccia di rete del router con la subnet *2001:1::* si possono osservare tutte la varie fasi necessarie per l'autoconfigurazione.

```
1 0.000000  ::                  ff02::1:ff00:2        ICMPv6     90 Neighbor Solicitation for 2001:1::200:ff:fe00:2 from 00:00:00:00:00:02
2 0.006000  ::                  ff02::1:ff00:2        ICMPv6     90 Neighbor Solicitation for fe80::200:ff:fe00:2 from 00:00:00:00:00:02
3 0.008144  ::                  ff02::1:ff00:3        ICMPv6     90 Neighbor Solicitation for fe80::200:ff:fe00:3 from 00:00:00:00:00:03
4 0.010291  ::                  ff02::1:ff00:1        ICMPv6     90 Neighbor Solicitation for fe80::200:ff:fe00:1 from 00:00:00:00:00:01
```

DAD 1 - in una prima fase si hanno i *neighbor solicitation*. Il router richiede fin da subito link local e link global alle righe 1, 2 (si noti come in questo caso la richiesta per il global appaia prima di quella del local) mentre i nodi host eseguono il neighbor solicitation per il link local address (righe 3-4). Il source address di ogni nodo in questa fase non è specificato proprio perché i nodi devono ancora settare il proprio indirizzo. L'indirizzo che viene richiesto è per ogni nodo dato dalla combinazione del prefix per gli indirizzi link local *fe80::/64* e l'indirizzo MAC dell'interfaccia di rete;

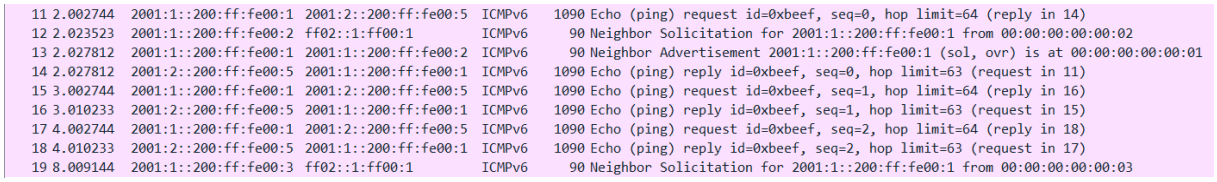
```
5 0.999000  fe80::200:ff:fe00:2  ff02::1              ICMPv6    114 Router Advertisement from 00:00:00:00:00:02
```

RA - il router periodicamente invia messaggi di tipo *router advertisement* verso l'indirizzo *ff02::1* che indica l'*all nodes multicast*. Questi messaggi hanno come obiettivo quello di permettere ai nodi host di conoscere il network prefix della loro rete;



6	1.007326	::	ff02::1:ff00:3	ICMPv6	90	Neighbor Solicitation for 2001:1::200:ff:fe00:3 from 00:00:00:00:00:03
7	1.010118	fe80::200:ff:fe00:3	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:03
8	1.012428	fe80::200:ff:fe00:1	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:01
9	1.013118	fe80::200:ff:fe00:2	ff02::1	ICMPv6	114	Router Advertisement from 00:00:00:00:00:02
10	1.014952	::	ff02::1:ff00:1	ICMPv6	90	Neighbor Solicitation for 2001:1::200:ff:fe00:1 from 00:00:00:00:00:01

DAD 2 - in questa fase si possono identificare i messaggi di *neighbor solicitation* per gli indirizzi link global. Sostanzialmente il processo è analogo al procedimento per il settaggio del link local. I vari nodi host mandano un messaggio all'indirizzo ottenuto dalla combinazione del network prefix ricevuto tramite i messaggi precedenti (in questo caso *2001::1*) e del loro link local address. Nell'immagine precedente si possono identificare anche i messaggi di *router solicitation* inviati dagli hosts verso l'indirizzo *ff02::2* (all routers multicast). Questi messaggi servono per sollecitare il router a mandare un router advertisement per poter configurare il loro link global address;



11	2.002744	2001:1::200:ff:fe00:1	2001:2::200:ff:fe00:5	ICMPv6	1090	Echo (ping) request id=0xbeef, seq=0, hop limit=64 (reply in 14)
12	2.023523	2001:1::200:ff:fe00:2	ff02::1:ff00:1	ICMPv6	90	Neighbor Solicitation for 2001:1::200:ff:fe00:1 from 00:00:00:00:00:02
13	2.027812	2001:1::200:ff:fe00:1	2001:1::200:ff:fe00:2	ICMPv6	90	Neighbor Advertisement 2001:1::200:ff:fe00:1 (sol, ovr) is at 00:00:00:00:00:01
14	2.027812	2001:2::200:ff:fe00:5	2001:1::200:ff:fe00:1	ICMPv6	1090	Echo (ping) reply id=0xbeef, seq=0, hop limit=63 (request in 11)
15	3.002744	2001:1::200:ff:fe00:1	2001:2::200:ff:fe00:5	ICMPv6	1090	Echo (ping) request id=0xbeef, seq=1, hop limit=64 (reply in 16)
16	3.010233	2001:2::200:ff:fe00:5	2001:1::200:ff:fe00:1	ICMPv6	1090	Echo (ping) reply id=0xbeef, seq=1, hop limit=63 (request in 15)
17	4.002744	2001:1::200:ff:fe00:1	2001:2::200:ff:fe00:5	ICMPv6	1090	Echo (ping) request id=0xbeef, seq=2, hop limit=64 (reply in 18)
18	4.010233	2001:2::200:ff:fe00:5	2001:1::200:ff:fe00:1	ICMPv6	1090	Echo (ping) reply id=0xbeef, seq=2, hop limit=63 (request in 17)
19	8.009144	2001:1::200:ff:fe00:3	ff02::1:ff00:1	ICMPv6	90	Neighbor Solicitation for 2001:1::200:ff:fe00:1 from 00:00:00:00:00:03

PING - in quest'ultima fase si possono riconoscere i messaggi di ping inviati dal nodo H_1 verso H_2 (request) e le successive *reply*. All'interno di questa fase si possono riconoscere anche messaggi di *neighbor solicitation* che a differenza di quelli che si possono vedere nella fase DAD servono per far conoscere ai nodi di una stessa subnet l'indirizzo IP global in modo reciproco (linee 12-13). Questo permetterà di mandare messaggi in modo diretto fra i nodi senza passare dal router.

C'è da notare come i pacchetti relativi al ping di $H3$ verso H_1 non siano visibili dall'interfaccia del router nonostante nei file *h1_interface.pcap* e *h3_interface.pcap* (relativi all'interfacce di rete di H_1 e H_3) siano visibili e il processo sia andato a buon fine.

Questo è un comportamento normale dal momento che i due nodi trovandosi nella stessa subnet non hanno bisogno del router per instradare i pacchetti da uno verso l'altro. Il router come visto sopra è però essenziale quando un nodo deve interfacciarsi verso l'esterno della rete.

Infine il file *R_subnet_2.pcap* permette di vedere l'interfaccia di rete con la subnet *2001:2::*. Anche in questo file is possono ritrovare tutti i pacchetti di cui si è discusso sopra con una struttura semplificata. La scelta di analizzare l'interfaccia di rete del router con la prima subnet discende proprio da questa osservazione.