Rapport TP2 - IOT

1. Compilation

Pour commencer on compile les firmwares FTD et MTD sur le channel 12 avec le PANID `0x12f0/`

```
$ iotlab-auth -u iot2023stras6
$ git clone https://github.com/RIOT-OS/RIOT.git
$ cd RIOT
$ git checkout 2022.07
$ cd examples/openthread
$ source /opt/riot.source
$ make BOARD=iotlab-m3 OPENTHREAD_CHANNEL=12 OPENTHREAD_PANID=0x12f0 OPENTHREAD_TYPE=ftd
$ cp bin/iotlab-m3/openthread.elf openthread-ftd.elf
$ source /opt/riot.source
$ make BOARD=iotlab-m3 OPENTHREAD_CHANNEL=12 OPENTHREAD_PANID=0x12f0 OPENTHREAD_TYPE=mtd
$ cp bin/iotlab-m3/openthread.elf openthread-mtd.elf
```

Les deux binaires sont maintenant trouvable dans `~/RIOT/examples/iotlab-m3`.

2. Déploiement du réseau

Une fois les 5 noeuds déployés avec les bon firmawares (2 sur FTD et 3 sur MTD), on peut leur définir leur masterkey :

```
$ serial_aggregator
$ ifconfig down
$ thread stop
$ masterkey 00cafedeca00deadbeaf00deafbee000
$ ifconfig up
$ thread start
```

On va pouvoir sniffer le réseau afin de comprendre le fonctionnement de :

1. l'election du leader
2. l'election des parents sur les noeuds End-Device
3. l'attribution des adresses (RLOC, IPv6)

Pour cela :

```
$ ssh iot-lab sniffer_aggregator -i 382202 -r -d -o - | wireshark -k -i -
```

On renseigne égallement la clé de décryptage `masterkey : 00cafedeca00deadbeaf00deafbee000` dans wireshark.

1. Élection du leader

Dès leur allumage, les noeuds de type FTD (chez nous`m3-12` et `m3-14`) commencent à broadcast leur `Link request` pour s'annoncer à l'autre FTD.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	m3-12	ip6-allrouters	MLE	61	Link Request
2	0.001428	m3-12	ip6-allrouters	MLE	61	Link Request
3	0.006645	m3-14	ip6-allrouters	MLE	61	Link Request
4	0.000407	m3-12	ip6-allrouters	MLE	61	Link Request
5	0.018353	m3-12	ip6-allrouters	MLE	61	Link Request
6	0.005625	m3-14	ip6-allrouters	MLE	61	Link Request
7	0.023570	m3-14	ip6-allrouters	MLE	61	Link Request
8	-0.000336	m3-12	ip6-allrouters	MLE	61	Link Request
9	0.004882	m3-14	ip6-allrouters	MLE	61	Link Request
10	0.005218	m3-14	ip6-allrouters	MLE	61	Link Request

Chaque routeur leader (chez nous `m3-14`) répond à l'autre FTD avec avec un `Link Request + Link Accept`.

227	6.159839	m3-14	m3-12	MLE	104	Link Accept and Request
228	6.161267	m3-14	m3-12	MLE	104	Link Accept and Request
229	6.160246	m3-14	m3-12	MLE	104	Link Accept and Request
230	6.178222	m3-14	m3-12	MLE	104	Link Accept and Request
231	6.159472	m3-14	m3-12	MLE	104	Link Accept and Request

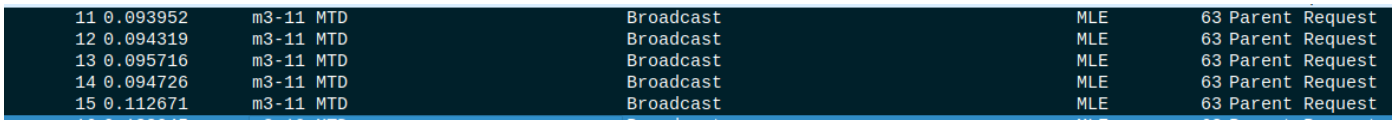
Pour finir, le routeur non leader (chez nous, `m3-12`) valide l'élection avec un `Link Accept`.

2. Élection des parents sur les noeuds End-Device

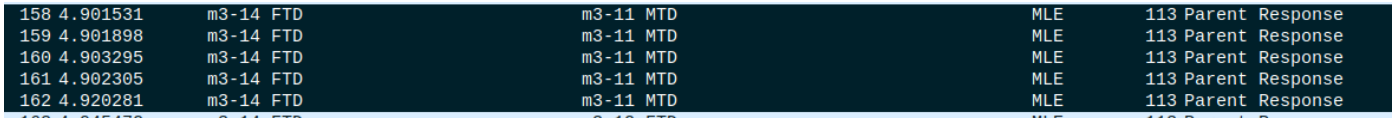
L'élection des parents est assez similaire à l'élection du leader.

Dès leur allumage, les noeuds de type MTD (chez nous `m3-10`,`m3-11` et `m3-13`) commencent à broadcast leur `Parent request` pour s'annoncer aux FTD.

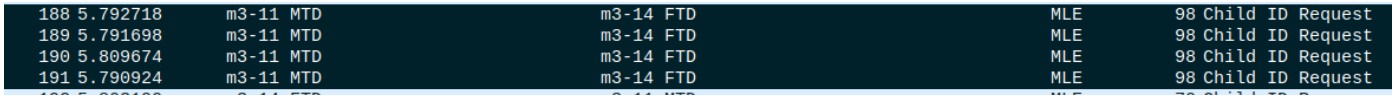
Dans les captures d'écran suivantes, nous verrons l'établissement du lien entre `m3-11` et le routeur `m3-14`.



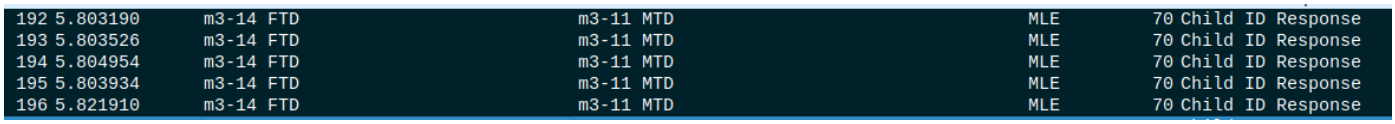
Ensuite, les routeurs ayant reçu la `Parent request` répondent au MTD avec un `Parent response`.



Ensuite, le noeud MTD renvoie au parent qu'il aura choisi un `Child ID request` pour récupérer un identifiant.

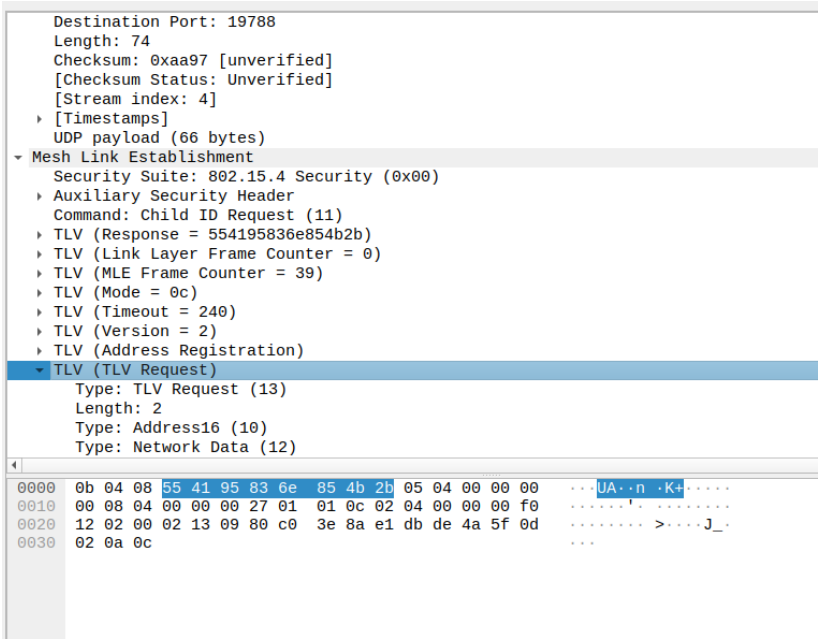


Et pour finir, le FTD élu comme parent répondra avec un `Child ID response` pour donner le nouvel identifiant à l'enfant.

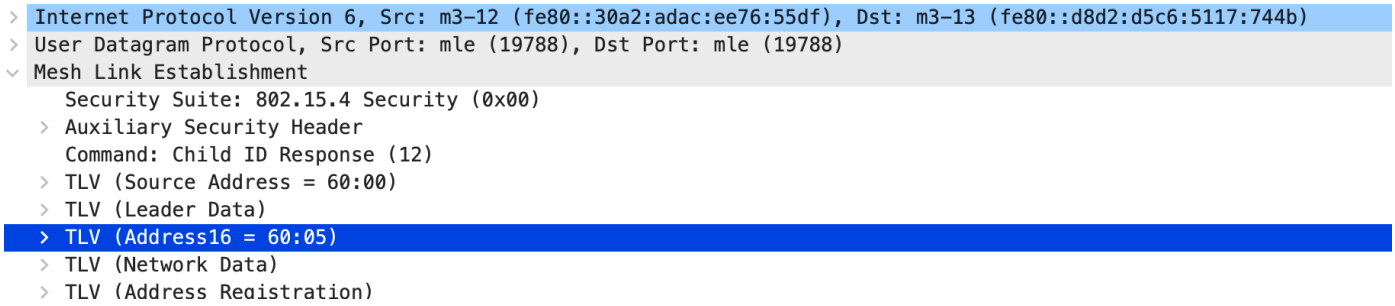


3. Attribution des adresses (RLOC, IPv6)

Lors de l'envoi du `Child ID Request` le MTD demande l'adresse `RLOC16` au routeur :

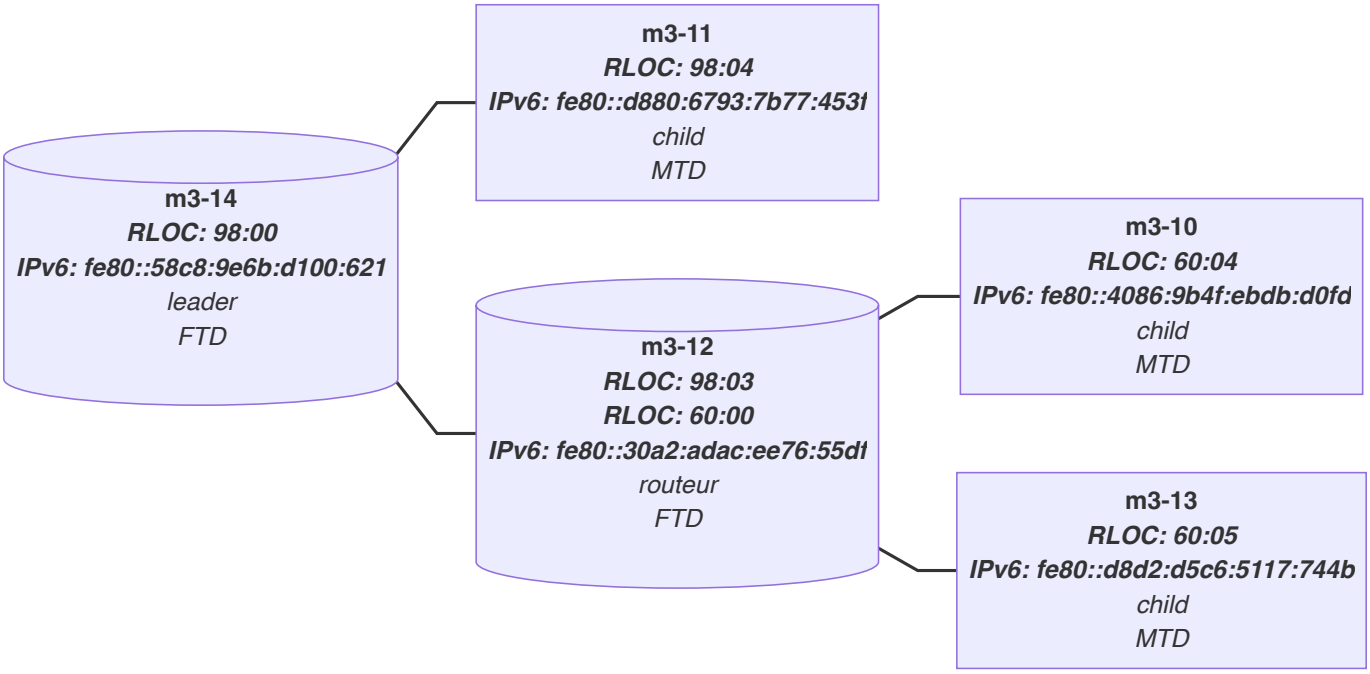


Lors de l'envoi du `Child ID Response` à son enfant, le FTD fourni une adresse `RLOC16` au MTD :



3. Plan du réseau

Grace aux commandes `status` et `child table` ainsi qu'avec les captures, on peut représenter le schéma de réseau suivant :



On peut constater que les noeuds `m3-10` et `m3-13` ne se sont pas appareillés avec le leader `m3-14`, mais avec le second routeur `m3-12`. Cela peut s'expliquer simplement : la connection entre `m3-14` et `m3-10`/`m3-13` est sans doute plus difficile, à cause des perturbations radios, ou de la distance entre eux par exemple. Et en effet, en faisant des tests avec `ping` on constate que les temps de réponses sont significativement plus lent entre `m3-14` et `m3-10`/`m3-13` que entre `m3-12` et `m3-10`/`m3-13`.

4. ICMPv6

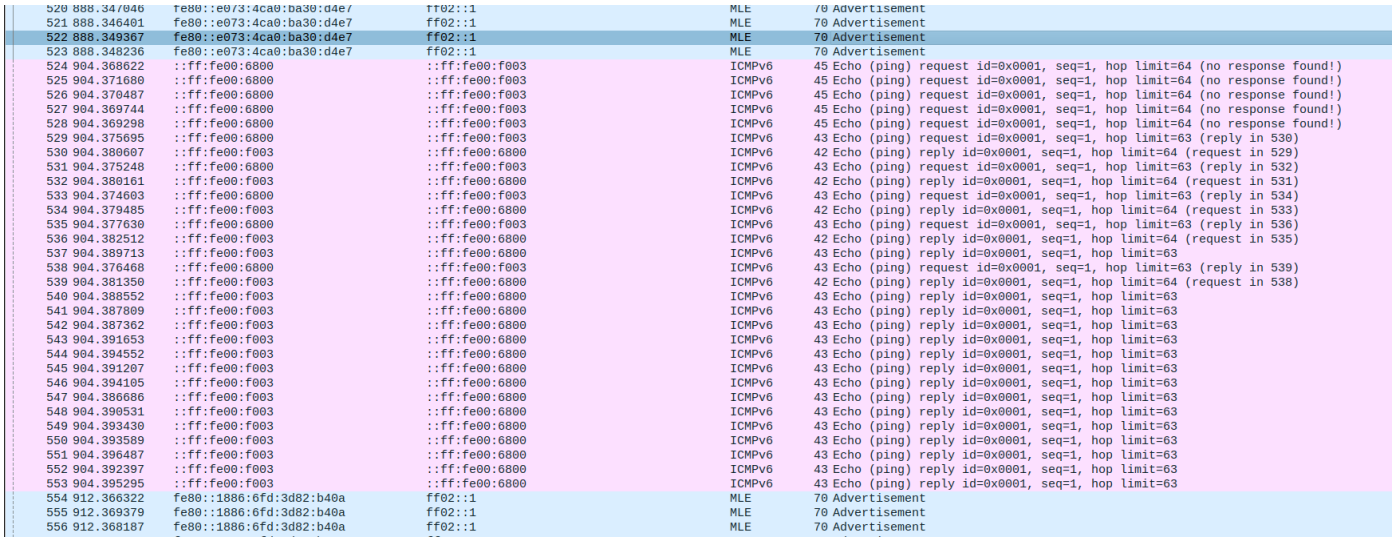
> **Note:** À partir de là, le travail à été effectué sur d'autres noeuds (les anciens étants occupés, ou défectueux), il est normal si les addresses ou les identifiants sont différents par rapport à la première partie du rapport.

Le connectivité entre nos noeuds est parfaitement fonctionnelle.

La commande suivante permet de pinger un de nos noeuds :

```
$ nc m3-10 20000
```

Le résultat est visible dans la capture wireshark :



520	888.347046	fe80::e073:4ca0:ba30:d4e7	ff02::1	MLE	70 Advertisement
521	888.346401	fe80::e073:4ca0:ba30:d4e7	ff02::1	MLE	70 Advertisement
522	888.349367	fe80::e073:4ca0:ba30:d4e7	ff02::1	MLE	70 Advertisement
523	888.348236	fe80::e073:4ca0:ba30:d4e7	ff02::1	MLE	70 Advertisement
524	904.368622	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	45 Echo (ping) request id=0x0001, seq=1, hop limit=64 (no response found!)
525	904.371680	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	45 Echo (ping) request id=0x0001, seq=1, hop limit=64 (no response found!)
526	904.370487	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	45 Echo (ping) request id=0x0001, seq=1, hop limit=64 (no response found!)
527	904.369744	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	45 Echo (ping) request id=0x0001, seq=1, hop limit=64 (no response found!)
528	904.369298	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	45 Echo (ping) request id=0x0001, seq=1, hop limit=64 (no response found!)
529	904.375695	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	43 Echo (ping) request id=0x0001, seq=1, hop limit=63 (reply in 530)
530	904.380607	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	42 Echo (ping) reply id=0x0001, seq=1, hop limit=64 (request in 529)
531	904.375248	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	43 Echo (ping) request id=0x0001, seq=1, hop limit=63 (reply in 532)
532	904.380161	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	42 Echo (ping) reply id=0x0001, seq=1, hop limit=64 (request in 531)
533	904.374603	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	43 Echo (ping) request id=0x0001, seq=1, hop limit=63 (reply in 534)
534	904.379485	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	42 Echo (ping) reply id=0x0001, seq=1, hop limit=64 (request in 533)
535	904.377630	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	43 Echo (ping) request id=0x0001, seq=1, hop limit=63 (reply in 536)
536	904.382512	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	42 Echo (ping) reply id=0x0001, seq=1, hop limit=64 (request in 535)
537	904.389713	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
538	904.376468	::ff:fe00:6800	::ff:fe00:f003	ICMPv6	43 Echo (ping) request id=0x0001, seq=1, hop limit=63 (reply in 539)
539	904.381350	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	42 Echo (ping) reply id=0x0001, seq=1, hop limit=64 (request in 538)
540	904.388552	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
541	904.387809	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
542	904.387362	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
543	904.391653	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
544	904.394552	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
545	904.391207	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
546	904.394105	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
547	904.386686	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
548	904.390531	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
549	904.393430	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
550	904.393589	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
551	904.396487	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
552	904.392397	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
553	904.395295	::ff:fe00:f003	::ff:fe00:6800	ICMPv6	43 Echo (ping) reply id=0x0001, seq=1, hop limit=63
554	912.366322	fe80::1886:6fd:3d82:b40a	ff02::1	MLE	70 Advertisement
555	912.369379	fe80::1886:6fd:3d82:b40a	ff02::1	MLE	70 Advertisement
556	912.368187	fe80::1886:6fd:3d82:b40a	ff02::1	MLE	70 Advertisement

5. Connectivité UDP (CoAP)

Pour commencer, on doit créer la ressource :

```
iot2023stras6@strasbourg:~$ nc m3-53 20000
coap start
coap start
Done
> coap resource test
coap resource test
Done
>
```

La seconde étape est de récuperer la ressouce :

```
iot2023stras6@strasbourg:~$ nc m3-56 20000
coap start
coap start
Done
> coap get fdde:ad00:beef:0:0:ff:fe00:6800 test
coap get fdde:ad00:beef:0:0:ff:fe00:6800 test
Done
> coap response from fdde:ad00:beef:0:0:ff:fe00:6800 with payload: 30
```

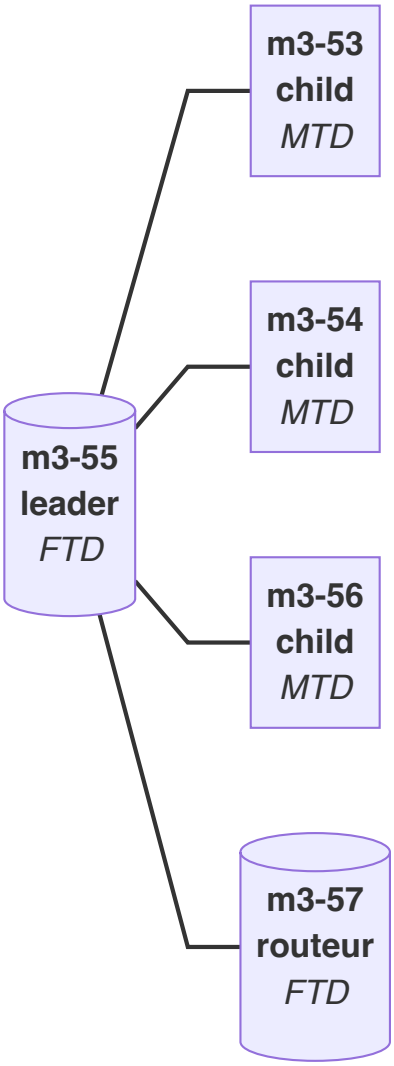
La capture wireshark nous montre que le transfert UDP CoAP c'est déroulé sans encombre :

909	1627.112251	fe80::e073:4ca0:ba30:d4e7	ff02::1	MLE	70 Advertisement
910	1635.694861	::ff:fe00:f002	::ff:fe00:6800	CoAP	43 NON, MID:36192, GET, TKN:0b 4d, /test
911	1635.692701	::ff:fe00:f002	::ff:fe00:6800	CoAP	43 NON, MID:36192, GET, TKN:0b 4d, /test [Retransmission]
912	1635.696245	::ff:fe00:f002	::ff:fe00:6800	CoAP	43 NON, MID:36192, GET, TKN:0b 4d, /test [Retransmission]
913	1635.698322	::ff:fe00:f002	::ff:fe00:6800	CoAP	43 NON, MID:36192, GET, TKN:0b 4d, /test [Retransmission]
914	1635.702899	::ff:fe00:f002	::ff:fe00:6800	CoAP	44 NON, MID:36192, GET, TKN:0b 4d, /test [Retransmission]
915	1635.694109	::ff:fe00:f002	::ff:fe00:6800	CoAP	43 NON, MID:36192, GET, TKN:0b 4d, /test [Retransmission]
916	1635.698686	::ff:fe00:f002	::ff:fe00:6800	CoAP	44 NON, MID:36192, GET, TKN:0b 4d, /test [Retransmission]
917	1635.707261	::ff:fe00:6800	::ff:fe00:f002	CoAP	44 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
918	1635.699438	::ff:fe00:f002	::ff:fe00:6800	CoAP	44 NON, MID:36192, GET, TKN:0b 4d, /test [Retransmission]
919	1635.708013	::ff:fe00:6800	::ff:fe00:f002	CoAP	44 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
920	1635.697279	::ff:fe00:f002	::ff:fe00:6800	CoAP	44 NON, MID:36192, GET, TKN:0b 4d, /test [Retransmission]
921	1635.700822	::ff:fe00:f002	::ff:fe00:6800	CoAP	44 NON, MID:36192, GET, TKN:0b 4d, /test [Retransmission]
922	1635.709397	::ff:fe00:6800	::ff:fe00:f002	CoAP	44 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
923	1635.714706	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
924	1635.705853	::ff:fe00:6800	::ff:fe00:f002	CoAP	44 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
925	1635.711162	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
926	1635.711474	::ff:fe00:6800	::ff:fe00:f002	CoAP	44 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
927	1635.716783	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
928	1635.712570	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
929	1635.717330	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
930	1635.713322	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
931	1635.718082	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
932	1635.715923	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
933	1635.719466	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
934	1635.721543	::ff:fe00:6800	::ff:fe00:f002	CoAP	41 ACK, MID:36192, 2.05 Content, TKN:0b 4d, /test
935	1638.908417	fe80::94af:e4f3:1419:bca6	fe80::e073:4ca0:ba30:d4e7	MLE	82 Child Update Request

On constate aussi que les messages ne sont pas assez long pour nécessiter une fragmentation. Ils ont néanmoins été émis plusieurs fois.

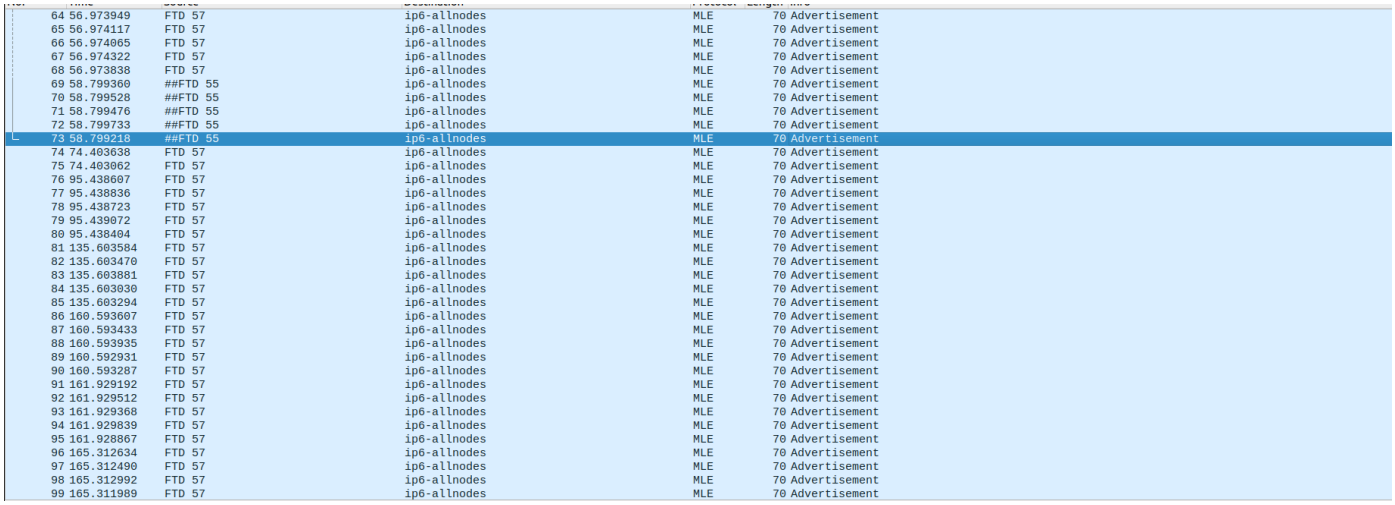
6. Changement de leader

Dans cette question, on considérera la structure suivante :



Pour commencer on éteint le noeud leader `m3-55`.

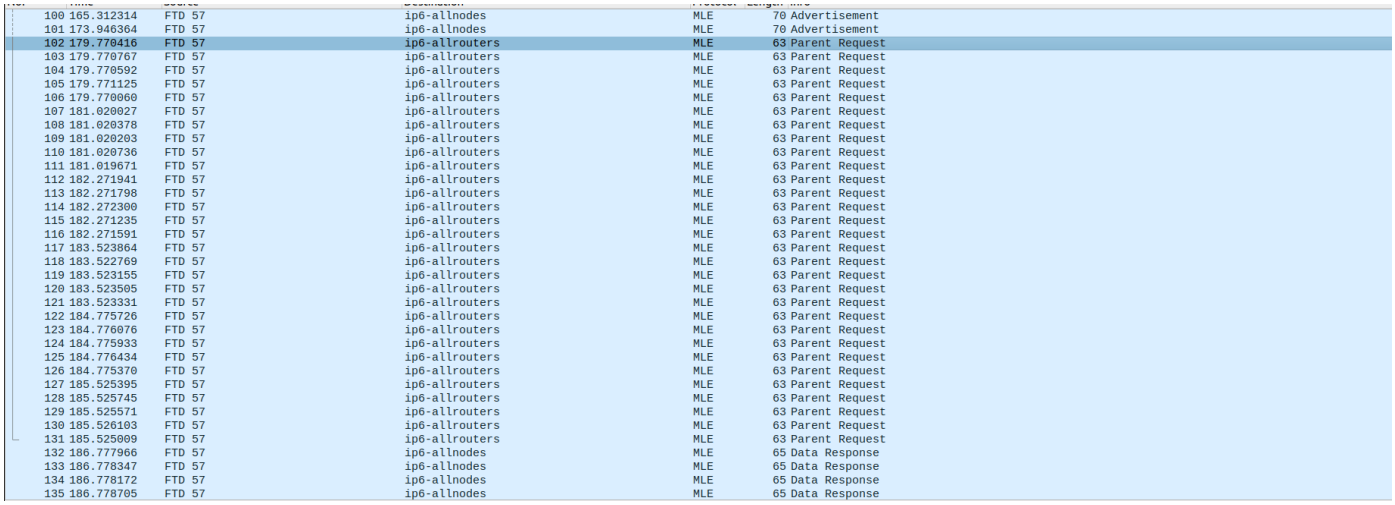
On remarque que le second FTD, le noeud `m3-57` envoie plusieurs message `MLE Advertissement` avant de détecter l'absence de leader.



64	56.973949	FTD 57	ip6-allnodes	MLE	70 Advertisement
65	56.974117	FTD 57	ip6-allnodes	MLE	70 Advertisement
66	56.974065	FTD 57	ip6-allnodes	MLE	70 Advertisement
67	56.974322	FTD 57	ip6-allnodes	MLE	70 Advertisement
68	56.973838	FTD 57	ip6-allnodes	MLE	70 Advertisement
69	58.799360	##FTD 55	ip6-allnodes	MLE	70 Advertisement
70	58.799528	##FTD 55	ip6-allnodes	MLE	70 Advertisement
71	58.799476	##FTD 55	ip6-allnodes	MLE	70 Advertisement
72	58.799733	##FTD 55	ip6-allnodes	MLE	70 Advertisement
73	58.799218	##FTD 55	ip6-allnodes	MLE	70 Advertisement
74	74.403638	FTD 57	ip6-allnodes	MLE	70 Advertisement
75	74.403062	FTD 57	ip6-allnodes	MLE	70 Advertisement
76	95.438607	FTD 57	ip6-allnodes	MLE	70 Advertisement
77	95.438836	FTD 57	ip6-allnodes	MLE	70 Advertisement
78	95.438723	FTD 57	ip6-allnodes	MLE	70 Advertisement
79	95.439072	FTD 57	ip6-allnodes	MLE	70 Advertisement
80	95.438404	FTD 57	ip6-allnodes	MLE	70 Advertisement
81	135.603584	FTD 57	ip6-allnodes	MLE	70 Advertisement
82	135.603470	FTD 57	ip6-allnodes	MLE	70 Advertisement
83	135.603881	FTD 57	ip6-allnodes	MLE	70 Advertisement
84	135.603030	FTD 57	ip6-allnodes	MLE	70 Advertisement
85	135.603294	FTD 57	ip6-allnodes	MLE	70 Advertisement
86	160.593607	FTD 57	ip6-allnodes	MLE	70 Advertisement
87	160.593433	FTD 57	ip6-allnodes	MLE	70 Advertisement
88	160.593935	FTD 57	ip6-allnodes	MLE	70 Advertisement
89	160.592931	FTD 57	ip6-allnodes	MLE	70 Advertisement
90	160.593287	FTD 57	ip6-allnodes	MLE	70 Advertisement
91	161.929192	FTD 57	ip6-allnodes	MLE	70 Advertisement
92	161.929512	FTD 57	ip6-allnodes	MLE	70 Advertisement
93	161.929368	FTD 57	ip6-allnodes	MLE	70 Advertisement
94	161.929839	FTD 57	ip6-allnodes	MLE	70 Advertisement
95	161.928867	FTD 57	ip6-allnodes	MLE	70 Advertisement
96	165.312634	FTD 57	ip6-allnodes	MLE	70 Advertisement
97	165.312490	FTD 57	ip6-allnodes	MLE	70 Advertisement
98	165.312992	FTD 57	ip6-allnodes	MLE	70 Advertisement
99	165.311989	FTD 57	ip6-allnodes	MLE	70 Advertisement

m3-55 est éteint à partir du paquet marqué.

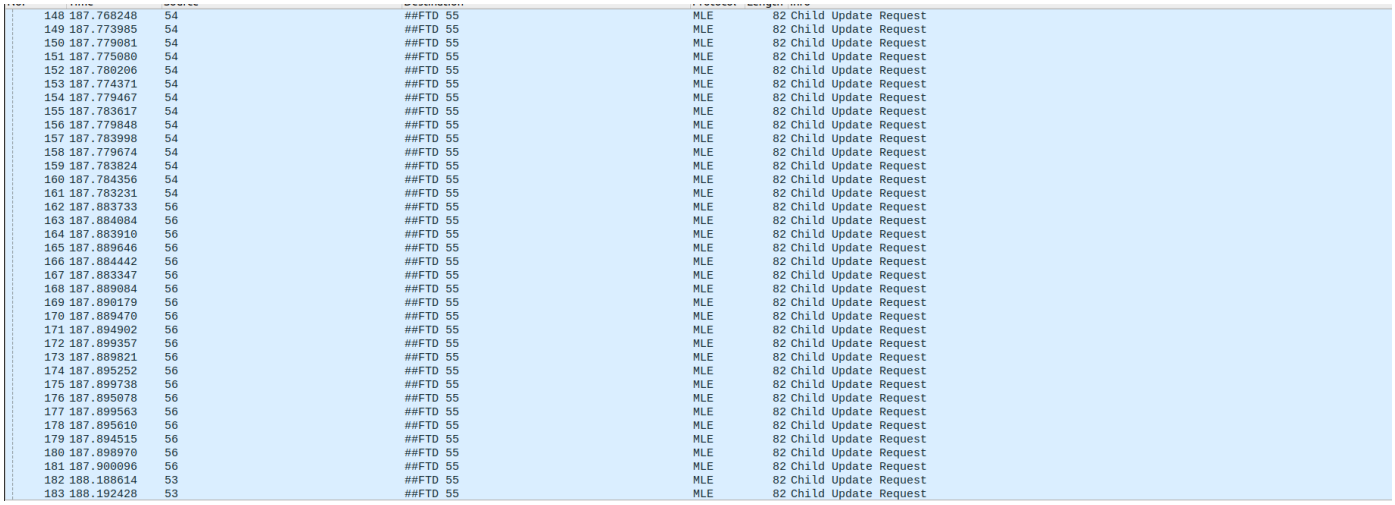
Dès que `m3-57` à détecter l'absence de leader, `m3-57` commence à broadcaster des `Parents request`.



100	165.312314	FTD 57	ip6-allnodes	MLE	70 Advertisement
101	173.946364	FTD 57	ip6-allnodes	MLE	70 Advertisement
102	179.770416	FTD 57	ip6-allrouters	MLE	63 Parent Request
103	179.770767	FTD 57	ip6-allrouters	MLE	63 Parent Request
104	179.770592	FTD 57	ip6-allrouters	MLE	63 Parent Request
105	179.771125	FTD 57	ip6-allrouters	MLE	63 Parent Request
106	179.770060	FTD 57	ip6-allrouters	MLE	63 Parent Request
107	181.020027	FTD 57	ip6-allrouters	MLE	63 Parent Request
108	181.020378	FTD 57	ip6-allrouters	MLE	63 Parent Request
109	181.020203	FTD 57	ip6-allrouters	MLE	63 Parent Request
110	181.020736	FTD 57	ip6-allrouters	MLE	63 Parent Request
111	181.019671	FTD 57	ip6-allrouters	MLE	63 Parent Request
112	182.271941	FTD 57	ip6-allrouters	MLE	63 Parent Request
113	182.271798	FTD 57	ip6-allrouters	MLE	63 Parent Request
114	182.272300	FTD 57	ip6-allrouters	MLE	63 Parent Request
115	182.271235	FTD 57	ip6-allrouters	MLE	63 Parent Request
116	182.271591	FTD 57	ip6-allrouters	MLE	63 Parent Request
117	183.523864	FTD 57	ip6-allrouters	MLE	63 Parent Request
118	183.522769	FTD 57	ip6-allrouters	MLE	63 Parent Request
119	183.523155	FTD 57	ip6-allrouters	MLE	63 Parent Request
120	183.523505	FTD 57	ip6-allrouters	MLE	63 Parent Request
121	183.523331	FTD 57	ip6-allrouters	MLE	63 Parent Request
122	184.775726	FTD 57	ip6-allrouters	MLE	63 Parent Request
123	184.776076	FTD 57	ip6-allrouters	MLE	63 Parent Request
124	184.775933	FTD 57	ip6-allrouters	MLE	63 Parent Request
125	184.776434	FTD 57	ip6-allrouters	MLE	63 Parent Request
126	184.775370	FTD 57	ip6-allrouters	MLE	63 Parent Request
127	185.525395	FTD 57	ip6-allrouters	MLE	63 Parent Request
128	185.525745	FTD 57	ip6-allrouters	MLE	63 Parent Request
129	185.525571	FTD 57	ip6-allrouters	MLE	63 Parent Request
130	185.526103	FTD 57	ip6-allrouters	MLE	63 Parent Request
131	185.525009	FTD 57	ip6-allrouters	MLE	63 Parent Request
132	186.777966	FTD 57	ip6-allnodes	MLE	65 Data Response
133	186.778347	FTD 57	ip6-allnodes	MLE	65 Data Response
134	186.778172	FTD 57	ip6-allnodes	MLE	65 Data Response
135	186.778705	FTD 57	ip6-allnodes	MLE	65 Data Response

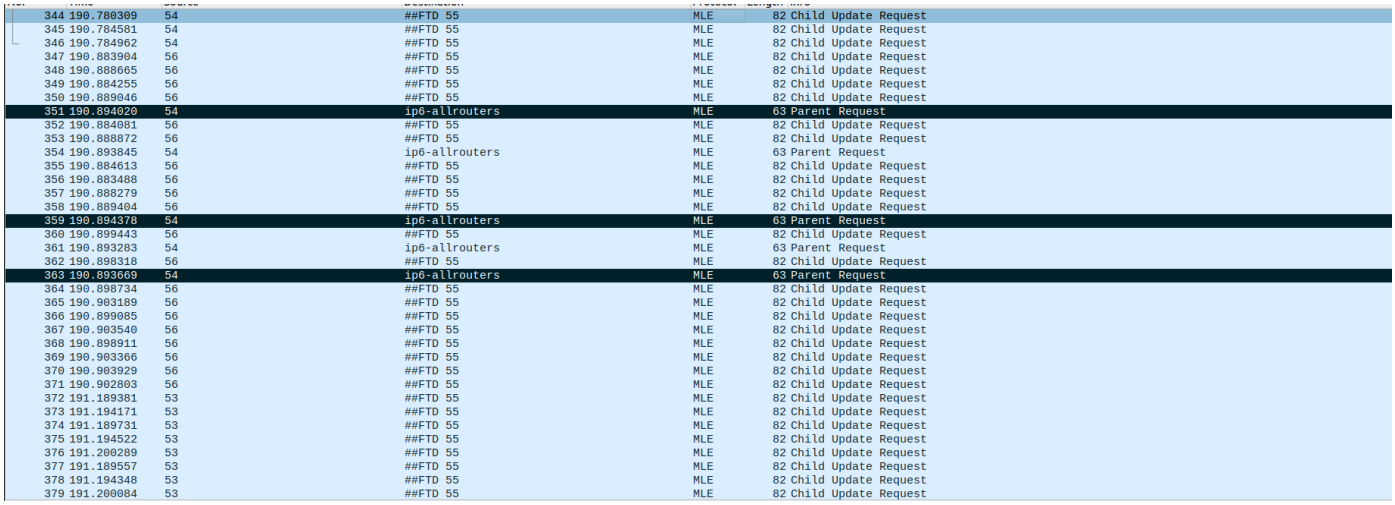
Les MTD childs (`m3-XX`, `m3-XX` et `m3-XX`) tentent de joindre l'ancien leader `m3-55`, sans succès.

Apres quelques Data Response, les childs essayent de joindre leur encien leader mais en vain en envoyant des `Child Update Request`



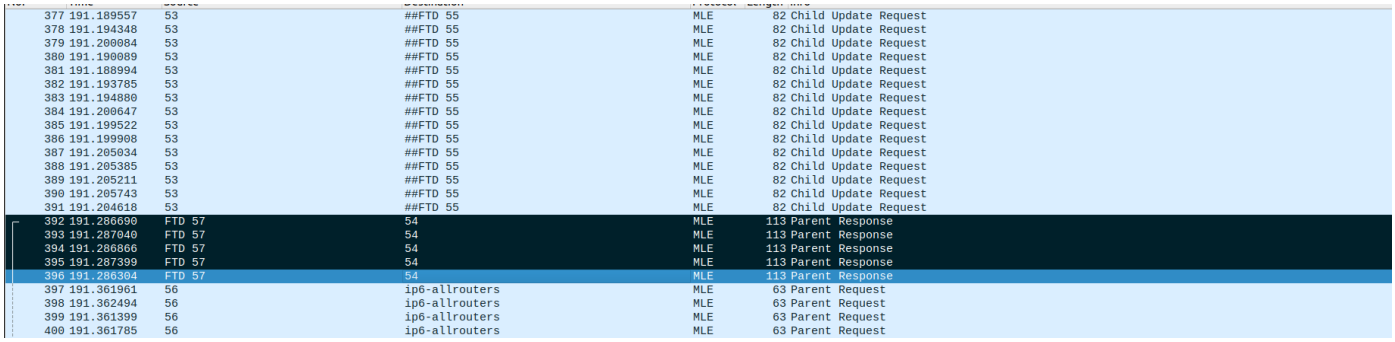
148	187.768248	54	##FTD 55	MLE	82 Child Update Request
149	187.773985	54	##FTD 55	MLE	82 Child Update Request
150	187.779081	54	##FTD 55	MLE	82 Child Update Request
151	187.775080	54	##FTD 55	MLE	82 Child Update Request
152	187.780206	54	##FTD 55	MLE	82 Child Update Request
153	187.774371	54	##FTD 55	MLE	82 Child Update Request
154	187.779467	54	##FTD 55	MLE	82 Child Update Request
155	187.783617	54	##FTD 55	MLE	82 Child Update Request
156	187.779848	54	##FTD 55	MLE	82 Child Update Request
157	187.783998	54	##FTD 55	MLE	82 Child Update Request
158	187.779674	54	##FTD 55	MLE	82 Child Update Request
159	187.783824	54	##FTD 55	MLE	82 Child Update Request
160	187.784356	54	##FTD 55	MLE	82 Child Update Request
161	187.783231	54	##FTD 55	MLE	82 Child Update Request
162	187.883733	56	##FTD 55	MLE	82 Child Update Request
163	187.884084	56	##FTD 55	MLE	82 Child Update Request
164	187.883910	56	##FTD 55	MLE	82 Child Update Request
165	187.889646	56	##FTD 55	MLE	82 Child Update Request
166	187.884442	56	##FTD 55	MLE	82 Child Update Request
167	187.883347	56	##FTD 55	MLE	82 Child Update Request
168	187.889084	56	##FTD 55	MLE	82 Child Update Request
169	187.890179	56	##FTD 55	MLE	82 Child Update Request
170	187.889470	56	##FTD 55	MLE	82 Child Update Request
171	187.894902	56	##FTD 55	MLE	82 Child Update Request
172	187.899357	56	##FTD 55	MLE	82 Child Update Request
173	187.889821	56	##FTD 55	MLE	82 Child Update Request
174	187.895252	56	##FTD 55	MLE	82 Child Update Request
175	187.899738	56	##FTD 55	MLE	82 Child Update Request
176	187.895078	56	##FTD 55	MLE	82 Child Update Request
177	187.899563	56	##FTD 55	MLE	82 Child Update Request
178	187.895610	56	##FTD 55	MLE	82 Child Update Request
179	187.894515	56	##FTD 55	MLE	82 Child Update Request
180	187.898970	56	##FTD 55	MLE	82 Child Update Request
181	187.900096	56	##FTD 55	MLE	82 Child Update Request
182	188.188614	53	##FTD 55	MLE	82 Child Update Request
183	188.192428	53	##FTD 55	MLE	82 Child Update Request

Les MTD ayant remarqué l'absence de leader, commencent à envoyer des `Parents Request`.



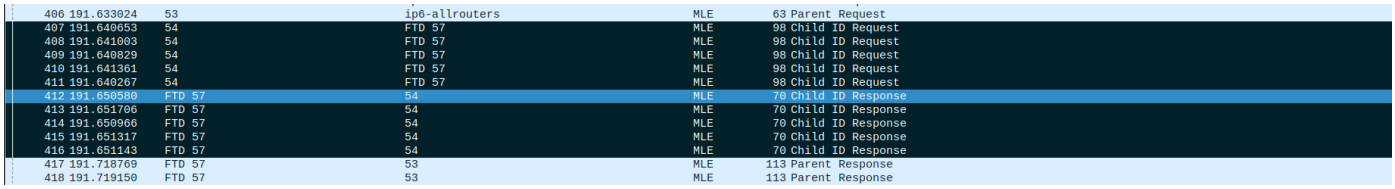
344	190.780309	54	##FTD 55	MLE	82 Child Update Request
345	190.784581	54	##FTD 55	MLE	82 Child Update Request
346	190.784962	54	##FTD 55	MLE	82 Child Update Request
347	190.883904	56	##FTD 55	MLE	82 Child Update Request
348	190.888665	56	##FTD 55	MLE	82 Child Update Request
349	190.884255	56	##FTD 55	MLE	82 Child Update Request
350	190.889046	56	##FTD 55	MLE	82 Child Update Request
351	190.894020	54	ip6-allrouters	MLE	63 Parent Request
352	190.884081	56	##FTD 55	MLE	82 Child Update Request
353	190.888872	56	##FTD 55	MLE	82 Child Update Request
354	190.893845	54	ip6-allrouters	MLE	63 Parent Request
355	190.884613	56	##FTD 55	MLE	82 Child Update Request
356	190.883488	56	##FTD 55	MLE	82 Child Update Request
357	190.888279	56	##FTD 55	MLE	82 Child Update Request
358	190.889404	56	##FTD 55	MLE	82 Child Update Request
359	190.894378	54	ip6-allrouters	MLE	63 Parent Request
360	190.893443	56	##FTD 55	MLE	82 Child Update Request
361	190.893283	54	ip6-allrouters	MLE	63 Parent Request
362	190.898318	56	##FTD 55	MLE	82 Child Update Request
363	190.893669	54	ip6-allrouters	MLE	63 Parent Request
364	190.898734	56	##FTD 55	MLE	82 Child Update Request
365	190.903189	56	##FTD 55	MLE	82 Child Update Request
366	190.899085	56	##FTD 55	MLE	82 Child Update Request
367	190.903540	56	##FTD 55	MLE	82 Child Update Request
368	190.898911	56	##FTD 55	MLE	82 Child Update Request
369	190.903366	56	##FTD 55	MLE	82 Child Update Request
370	190.903929	56	##FTD 55	MLE	82 Child Update Request
371	190.902803	56	##FTD 55	MLE	82 Child Update Request
372	191.189381	53	##FTD 55	MLE	82 Child Update Request
373	191.194171	53	##FTD 55	MLE	82 Child Update Request
374	191.189731	53	##FTD 55	MLE	82 Child Update Request
375	191.194522	53	##FTD 55	MLE	82 Child Update Request
376	191.200289	53	##FTD 55	MLE	82 Child Update Request
377	191.189557	53	##FTD 55	MLE	82 Child Update Request
378	191.194348	53	##FTD 55	MLE	82 Child Update Request
379	191.200084	53	##FTD 55	MLE	82 Child Update Request

À partir de là, cela se déroule comme une élection de parent classique. Les MTD vont donc recevoir un `Parent Response` de `m3-57` qui c'est autoproclamé leader.



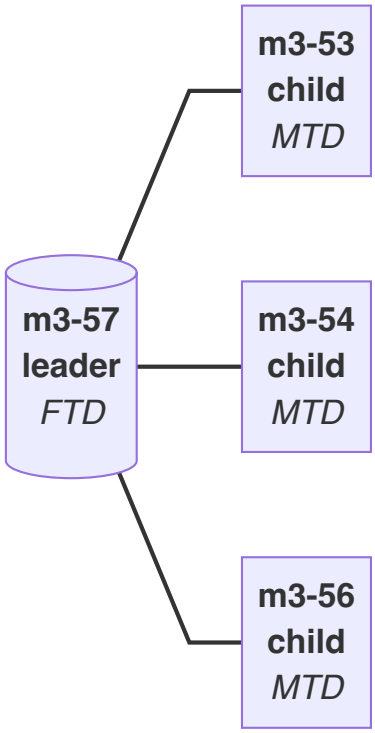
377	191.189557	53	##FTD 55	MLE	82 Child Update Request
378	191.194348	53	##FTD 55	MLE	82 Child Update Request
379	191.200084	53	##FTD 55	MLE	82 Child Update Request
380	191.190089	53	##FTD 55	MLE	82 Child Update Request
381	191.188994	53	##FTD 55	MLE	82 Child Update Request
382	191.193785	53	##FTD 55	MLE	82 Child Update Request
383	191.194880	53	##FTD 55	MLE	82 Child Update Request
384	191.200647	53	##FTD 55	MLE	82 Child Update Request
385	191.199522	53	##FTD 55	MLE	82 Child Update Request
386	191.199908	53	##FTD 55	MLE	82 Child Update Request
387	191.205034	53	##FTD 55	MLE	82 Child Update Request
388	191.205385	53	##FTD 55	MLE	82 Child Update Request
389	191.205211	53	##FTD 55	MLE	82 Child Update Request
390	191.205743	53	##FTD 55	MLE	82 Child Update Request
391	191.204618	53	##FTD 55	MLE	82 Child Update Request
392	191.286690	FTD 57	54	MLE	113 Parent Response
393	191.287040	FTD 57	54	MLE	113 Parent Response
394	191.286866	FTD 57	54	MLE	113 Parent Response
395	191.287399	FTD 57	54	MLE	113 Parent Response
396	191.286304	FTD 57	54	MLE	113 Parent Response
397	191.361961	56	ip6-allrouters	MLE	63 Parent Request
398	191.362494	56	ip6-allrouters	MLE	63 Parent Request
399	191.361399	56	ip6-allrouters	MLE	63 Parent Request
400	191.361785	56	ip6-allrouters	MLE	63 Parent Request

Et ensuite, comme dans une élection de parents classique, débute l'échange de `Child ID request` et de `Child ID response`



406	191.633024	53	ip6-allrouters	MLE	63 Parent Request
407	191.640653	54	FTD 57	MLE	98 Child ID Request
408	191.641003	54	FTD 57	MLE	98 Child ID Request
409	191.640829	54	FTD 57	MLE	98 Child ID Request
410	191.641361	54	FTD 57	MLE	98 Child ID Request
411	191.640267	54	FTD 57	MLE	98 Child ID Request
412	191.650580	FTD 57	54	MLE	70 Child ID Response
413	191.651706	FTD 57	54	MLE	70 Child ID Response
414	191.650966	FTD 57	54	MLE	70 Child ID Response
415	191.651317	FTD 57	54	MLE	70 Child ID Response
416	191.651143	FTD 57	54	MLE	70 Child ID Response
417	191.718769	FTD 57	53	MLE	113 Parent Response
418	191.719150	FTD 57	53	MLE	113 Parent Response

Assez naturellement, le réseau à maintenant cette topologie :


m3-53
child
MTD
m3-57
leader
FTD
m3-54
child
MTD
m3-56
child
MTD