

Amon - Tâche #37075

Scénario # 36946 (Terminé (Sprint)): AMON 2.10 : Erreur au reconfigure lors de l'ajout d'alias dans le config.eol

etude

08/09/2025 15:36 - Ludwig Seys

Statut:	Fermé	Début:	01/10/2022
Priorité:	Normal	Echéance:	
Assigné à:	Ludwig Seys	% réalisé:	100%
Version cible:	Livraison Cadoles - MEN 30/06/2025 (20)	Temps estimé:	0.00 heure
		Temps passé:	0.00 heure
Description			

Historique

#1 - 08/09/2025 15:36 - Ludwig Seys

- Statut changé de Nouveau à En cours

#2 - 09/09/2025 12:36 - Ludwig Seys

le problème vient sans doute d'un test dans reconfigure.

la fonction `_is_valid_ip_eth0` ne vérifie **que** la première IP remontée pour l'interface et la compare à `adresse_ip_eth0`. Dès qu'un alias est ajouté, l'ordre des adresses peut changer et l'alias peut passer en premier. Si la comparaison échoue pendant le timeout donné ça provoque le retour `False` et le message "Impossible d'obtenir une adresse IP".

Ce test n'est fait que sur `eth0`. (ligne1039 dans `reconfigure`)

```
eths = get_devices()
real_ip_eth0 = eths.get(nom_carte, ('169.254.0.1',))[0] # le problème est qu'ici on ne regarde que l'index 0
    if is_dhcp:
        if real_ip_eth0 != "169.254.0.1":
            return True
    else:
        if real_ip_eth0 == ip_eth0: # <-- exige égalité stricte avec l'IP principale
            return True
```

On compare l'IP courante après le restart de `networkd` (celles réellement assignées) à l'IP attendue (valeur Creole `adresse_ip_eth0`). Avec des alias sur `eth0`, il faut regarder toutes les IP assignées, pas seulement la première (entre autre parce que l'ordre peut changer).

Ce problème ne peut être révélé qu'en ajoutant un alias sur `eth0`.

La solution serait de modifier le code pour tester toutes les ip dans `eth0` (ip principale et alias!) ce qui ferait sens et respecterait la volonté première d'être sur qu'`eth0` à bien les ip données dans `genconfig`.

l'application de la configuration en passant par `netplan` ne provoque par d'erreur :

```
root@amon:~# netplan --debug generate
echo $?
DEBUG:command generate: running ['/usr/libexec/netplan/generate']
** (generate:119607): DEBUG: 12:14:33.913: starting new processing pass
** (generate:119607): DEBUG: 12:14:33.913: starting new processing pass
** (generate:119607): DEBUG: 12:14:33.913: enp4s0: adding new route
** (generate:119607): DEBUG: 12:14:33.913: enp4s0: adding new route
** (generate:119607): DEBUG: 12:14:33.913: enp4s0: adding new route
** (generate:119607): DEBUG: 12:14:33.913: enp5s0: adding new route
** (generate:119607): DEBUG: 12:14:33.913: enp6s0: adding new route
** (generate:119607): DEBUG: 12:14:33.913: enp7s0: adding new route
** (generate:119607): DEBUG: 12:14:33.913: We have some netdefs, pass them through a final round of validation
** (generate:119607): DEBUG: 12:14:33.913: enp6s0: setting default backend to 1
** (generate:119607): DEBUG: 12:14:33.913: Configuration is valid
** (generate:119607): DEBUG: 12:14:33.913: vlan21: setting default backend to 1
```

```

** (generate:119607): DEBUG: 12:14:33.913: Configuration is valid
** (generate:119607): DEBUG: 12:14:33.913: enp7s0: setting default backend to 1
** (generate:119607): DEBUG: 12:14:33.913: Configuration is valid
** (generate:119607): DEBUG: 12:14:33.913: enp4s0: setting default backend to 1
** (generate:119607): DEBUG: 12:14:33.913: Configuration is valid
** (generate:119607): DEBUG: 12:14:33.913: enp5s0: setting default backend to 1
** (generate:119607): DEBUG: 12:14:33.913: Configuration is valid
** (generate:119607): DEBUG: 12:14:33.913: vlan22: setting default backend to 1
** (generate:119607): DEBUG: 12:14:33.913: Configuration is valid
** (generate:119607): DEBUG: 12:14:33.913: Generating output files..
** (generate:119607): DEBUG: 12:14:33.914: Open vSwitch: definition enp4s0 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: NetworkManager: definition enp4s0 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: Open vSwitch: definition enp5s0 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: NetworkManager: definition enp5s0 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: Open vSwitch: definition enp6s0 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: NetworkManager: definition enp6s0 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: Open vSwitch: definition enp7s0 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: NetworkManager: definition enp7s0 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: Open vSwitch: definition vlan21 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: NetworkManager: definition vlan21 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: Open vSwitch: definition vlan22 is not for us (backend 1)
** (generate:119607): DEBUG: 12:14:33.914: NetworkManager: definition vlan22 is not for us (backend 1)
0

```

Dans ce cas en forçant sans passer par reconfigure les ip sont bien prises en charge !

```

2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:00:c0:a8:00:67 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.31/24 brd 192.168.0.255 scope global enp4s0
        valid_lft forever preferred_lft forever
    inet 192.168.0.60/26 brd 192.168.0.63 scope global enp4s0
        valid_lft forever preferred_lft forever

```

l'ordre est pas forcément linéaire,

```
ip -4 addr show dev eth0 | awk '/inet /{print NR: "$2}"'
```

1. ou

```
networkctl status eth0 | sed -n '/Address:./,/^$/p'
```

#3 - 09/09/2025 15:52 - Ludwig Seys

problème trouvé !

get_devices() ne donne qu'un seul triplet (ipv4, netmask, broadcast), et si il y a un alias il donne l'alias avant de donner l'ip "principal"

```
=> print(f"[DEBUG] nom_carte={nom_carte} ip_eth0={ip_eth0} get_devices()[iface]={eths.get(nom_carte)}")
```

```
renvoi : [DEBUG] nom_carte=enp4s0 ip_eth0=192.168.0.31 get_devices()[iface]=('192.168.0.60', '255.255.255.192', '192.168.0.63')
```

<

Pour les ip suivante :

```
root@amon:~# CreoleGet --list | grep eth0
activer_bonding_eth0="non"
activer_dns_eth0="non"
activer_promiscuous_eth0="non"
admin_eth0="oui"
adresse_broadcast_eth0="192.168.0.255"
adresse_ip_eth0="192.168.0.31"
adresse_netmask_eth0="255.255.255.0"
adresse_network_eth0="192.168.0.0"
alias_eth0="oui"
alias_ip_eth0="192.168.0.60"
alias_broadcast_eth0="192.168.0.63"
alias_netmask_eth0="255.255.255.192"
alias_network_eth0="192.168.0.0"
```

la première solution pourrait être de ne plus utiliser get_devices() => process.system_out(['ip', '-o', '-4', 'addr', 'show', 'dev', nom_carte]) ??

test en cours

#4 - 09/09/2025 16:28 - Ludwig Seys

- Statut changé de *En cours* à *À valider*

#5 - 23/09/2025 10:38 - Emmanuel GARETTE

- % réalisé changé de 0 à 100

#6 - 23/09/2025 11:07 - Emmanuel GARETTE

- Statut changé de *À valider* à *Résolu*

#7 - 26/09/2025 15:24 - Joël Cuissinat

- Statut changé de *Résolu* à *Fermé*

- Restant à faire (heures) mis à 0.0