

Création d'un serveur HBLink et son moniteur Web.

version du 31 janvier 2020

Base Linux Debian10 sur OVH

<https://www.ovh.com/fr/vps/vps-ssd.xml>



Table des matières :

- Quel intérêt de construire son propre serveur HBLink ?
- Organisation et fonctionnement du serveur
- Création d'un serveur HBLink et son moniteur Web
 - o Installation des prérequis
 - o Modification du fichier de configuration du serveur
 - o Modification du fichier des règles de fonctionnement du serveur
 - o Préparation du service et lancement automatique du serveur
 - o Installation du monitor WEB du serveur HBLink
 - o Préparation du service et lancement automatique du monitor
- Annexes
 - o Commandes Linux de gestion des services
 - o Gestion de la rotation des logs
 - o La personnalisation du tableau de bord du moniteur WEB
- Fonctions avancées
 - o Installation du service ECHO (Parrot)
 - o Interconnexion de 2 serveurs avec OpenBridge
 - o Interconnexion avec un serveur Brandmeister
- Paramètres de configurations d'un serveur HBLink3
 - o Configuration du serveur : **hblink.cfg**
 - o Configuration des règles : **rules.py**
 - o Explication des règles par l'exemple

Nous tenons à remercier pour leur participation directe ou indirecte :

- **Christopher WB4ULK** pour son tutoriel vidéo pour la création d'un serveur HBLink sous Linux :
<http://www.chrishoodblog.com/make-your-own-dmr-server>
- **IK5XMK** pour ses documentation quant à la configuration d'un serveur HBLink :
<http://www.grupporadiofirenze.net/2019/10/17/hblink3-e-i-primi-passi-per-creare-un-network-radio/>
<http://www.grupporadiofirenze.net/2019/10/23/hblink3-parte-seconda-hblink-cfg-e-rules-py/>
- **Hugues ON5HH** pour avoir été à l'origine des primo-configurations du serveur HBLink DMR Francophone.
- **Didier F4GIX** pour ses échanges techniques et conseils et en particulier pour la mise en place du service ECHO
<https://f0gix.skyrock.com>
- **Patrick F1FQN** pour la documentation, le support et la configuration et suivi technique du serveur.
- **Yannick F1TUV** pour avoir été l'initiateur de ce projet et fervent défenseur d'un DMR libre et simple pour tous.
<http://dmr-francophone.net>
- Nos fidèles testeurs : **Georges F5JRN, Michel F5MDR, Charles F4OGZ**

Support HBLink DMR-Francophone sur Telegram :

<http://t.me/hblinkTG900>

Support HBLink DMR-Francophone sur l'air :

137.74.195.150 port 62031 / passw0rd / TG7/Ts2 (Hotspots simplex)
137.74.195.150 port 62030 / passw0rd / TG7/Ts2 (Hotspots duplex)

Dashboard HBLink DMR francophone :

<http://137.74.195.150:8080>

Quel intérêt de construire son propre serveur HBLink ?

Vous connaissez sûrement le réseau Brandmeister pour l'avoir utilisé. Vous savez donc qu'il n'y a qu'un seul serveur MAITRE de raccordement par pays associé à un administrateur. Pour les autres OM's, il n'y a donc pas matière à participer activement et d'expérimenter et donc de comprendre le mode de gestion de ce type de réseau. Il faut bien le reconnaître, ne rester qu'un simple utilisateur peut être assez frustrant.

HBLink vous permet donc de fabriquer votre propre serveur, d'y connecter vos Hotspots, vos relais (problèmes avec les relais Hytera), d'apprendre à gérer les TG et leurs TS et ainsi pouvoir créer un réseau autonome pour vous seul, votre département, votre radio-club etc... Vous ne serez pas isolé pour autant car vous pourrez partager vos flux ou en récupérer, en vous connectant avec d'autres serveurs HBLink voire d'autres réseaux DMR (BM, IPSC2, XLX)

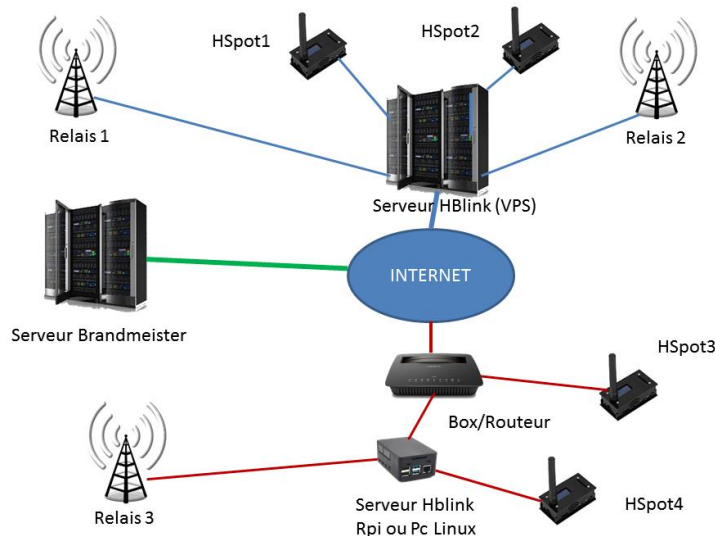
HBLink est un serveur gratuit, Open Source, ouvert sur un code en Python adaptable selon vos besoins. Vous ne dépendez de personne pour mener vos expérimentations. Attention toutefois à ne pas faire n'importe quoi quand vous vous raccordez à d'autres réseaux ?

Illustrons ceci par un exemple :

Le réseau **HBLink 1 (bleu)** est hébergé dans le "Cloud" sur un serveur privé virtuel (VPS). 2 hotspots et 2 relais MMDVM y sont raccordés.

Le réseau **HBLink 2 (rouge)** est hébergé sur un PC ou sur un Raspberry au domicile d'un OM. 1 hotspot extérieur, 1 hotspot intérieur et un relais MMDVM y sont raccordés.

1 serveur Brandmeister ou IPsc2 (facultatif) qui pourra éventuellement fournir des flux * pour cela, il vous faudra contacter le service support des réseaux auxquels vous voulez vous raccorder.



Chaque réseau peut faire le choix d'être indépendant ou se raccorder sur un autre réseau HBLink ou un autre réseau DMR.

Organisation et fonctionnement du serveur

Le fonctionnement du serveur ne repose essentiellement que sur la configuration de 2 fichiers principaux qui sont :

- **hblink.cfg**
- **rules.py**

Hormis les paramètres généraux, le fichier **hblink.cfg** permet principalement d'organiser des zones d'hébergements. Par défaut, **hblink.cfg** ne comporte qu'une seule zone nommée « Master-1 ». Vous pouvez créer autant de zones que vous le désirez.

Par exemple vous pouvez créer:

- 1 zone commune pour les Hotspots simplex
- 1 zone commune pour les Hotspots duplex
- 1 zone commune pour l'ensemble des relais ou une zone dédiée pour chaque relais.
- Des zones dédiées pour des Hotspots privilégiés

Toutefois, sachez que **tous les éléments d'une même zone partageront et donc écouteront le même trafic radio** (broadcast entre tous).

Chaque zone créée occupe et utilise un port UDP unique.

Afin d'éviter cela, il serait souhaitable de pouvoir créer autant de zones isolées que d'éléments connectés mais la configuration peut devenir assez fastidieuse (copié-collé de zones et adaptation d'un port UDP dédié...). A cause de ce manque de souplesse, un serveur HBLink n'est donc adapté qu'à la création de petits réseaux sauf si vous n'utilisez qu'une seule zone de broadcast et que vous mettez tout le monde dedans sur un seul port qui est par défaut le **62031** avec pour mot de passe "**passw0rd**".

Pour revenir à notre exemple :

- Les Hotspots simplex se partageront une même zone sur le port 62031/passw0rd
- Les Hotspots duplex se partageront une même zone sur le port 62030/passw0rd
- Les relais se partageront une même zone sur le port 55000/passw0rd

Ou bien : Chaque relais possèdera sa zone dédiée :

Relais 1 : 55001/Password

Relais 2 : 55002/passwd

Relais 3 : 55003/Drowssap

...

Avec des Hostspots particuliers sur des zones dédiées :

HotspotVIP 1 : 54001/motdepasse

Hotspot VIP2 : 54002/essapedtom

Pour chaque zone, il est possible de placer des autorisations et/ou des interdictions. A savoir, qui a le droit de se connecter, sur quel TG's, sur quel TS... C'est dans la zone ACL de chaque zone qu'il vous est possible de paramétrer tout ceci.

Création d'un serveur HBLink et son moniteur Web

Base linux Debian10 sur serveur VPS OVH

<https://www.ovh.com/fr/vps/vps-ssd.xml>

Se connecter en **SSH** à son serveur avec le mot de passe root, qu'OVH renvoie par mail quand le serveur Linux a achevé sa construction.

```
ssh root@[IP du serveur]
```

La première chose à faire est de changer le mot de passe par défaut de l'utilisateur root qui a été généré aléatoirement lors de la conception du serveur Debian 10.

```
#passwd
```

On entre 2 fois le nouveau mot de passe.

Puis ensuite, sécurisez à minima votre serveur :

<https://docs.ovh.com/fr/vps/conseils-securisation-vps/>

1 - Installation des prérequis

On se place dans le répertoire /opt

```
cd /opt
apt-get update
apt-get upgrade
apt-get install git
wget http://bootstrap.pypa.io/get-pip.py
python3 get-pip.py Si cela ne fonctionne pas utilisez python get-pip.py.
(Pour la suite du tuto vous devrez utiliser python3)
apt-get install python3-twisted
git clone https://github.com/n0mjs710/dmr_utils3.git
git clone https://github.com/n0mjs710/HBLink3.git
```

On se place dans le répertoire /opt/dmr_utils3

```
cd dmr_utils3
chmod +x install.sh
./install.sh
```

On se place dans le répertoire de configuration de HBLink3

2 - Modification du fichier de configuration du serveur HBLink

```
cd /opt/HBlink3
cp hblink-SAMPLE.cfg hblink.cfg
nano hblink.cfg
```

Avec nano et la combinaison de touches **Control + W** (fonction "rechercher").
Modifiez les sections suivantes dans le fichier HbLink3.cfg

```
[LOGGER]
LOG_FILE: /var/log/hblink.log
LOG_HANDLERS: console-timed,file-timed
LOG_LEVEL: INFO
LOG_NAME: HBlink
```

```
[OBP-1]
MODE: OPENBRIDGE
ENABLED: False
```

```
[MASTER-1]
MODE: MASTER
ENABLED: True
REPEAT: True
MAX_PEERS: 10
EXPORT_AMBE: False
IP:
PORT: 62031
PASSPHRASE: passw0rd
GROUP_HANGTIME: 5
USE_ACL: True
REG_ACL: DENY:1
SUB_ACL: DENY:1
TGID_TS1_ACL: PERMIT:ALL
TGID_TS2_ACL: PERMIT:ALL
```

```
[REPEATER-1] MODE: PEER
ENABLED: False
```

...

```
[XLX-1]
MODE: XLXPEER
ENABLED: False
```



DMR
Francophone

La section [OBP-1]

Est la section OpenBridge qui sert à raccorder votre serveur à d'autres serveurs dans le but de partager des flux de TG. Vous pouvez créer autant de zones OBP par simple "copier-coller" en y adaptant les ports UDP et les mots de passe en accord avec l'administrateur du serveur sur le lequel ce pont sera créé.

La section [MASTER-1]

Accueille vos « invités » tels que les Hotspots, Raspberry Pi Dvswitch, Relais MMDVM... Vous pouvez créer autant de zones MASTER par simple « copié-collé », toujours en adaptant un nouveau port UDP à chaque zone MASTER ainsi que le mots de passe.

La section [REPEATER- 1]

Permet de créer des "Hotspots virtuels". Ces "Hotspots virtuels" se comporteront, vu du serveur de raccordement comme un classique hotspot MMDVM physique. Cela permet donc de vous connecter aux TG's d'autres serveurs et de les mettre à disposition sur le vôtre.

Toutefois, veuillez lire les articles suivants :

<https://brandmeister-dmr.fr/2019/08/31/le-protocole-mmdvm-ne-doit-pas-etre-utilise-pour-faire-des-interconnexions-entre-brandmeister-et-dautres-reseaux-amateur-digitaux/>

et

<https://brandmeister-dmr.fr/2019/01/02/interconnexions-des-talkgroups-brandmeister-dmr-avec-dautres-reseaux/>

La section [XLX]

Permet de raccorder votre serveur HBLink à des serveurs de type XLX.

Important : Pour chaque zone (section) que vous aurez créé, il vous faudra créer des règles de routage qui permettront de distribuer vos TG / TS dans les zones que vous voudrez les affecter.

Tout cela est détaillé en fin de document.

3 - Modification du fichier des règles de fonctionnement du serveur HBLink.

```
cp rules_SAMPLE.py rules.py  
nano rules.py
```

Avec nano et la combinaison des touches Control + W (fonction « recherchez ») modifier la section « BRIDGES = » dans le fichier **rules.py** afin de ne laissez subsister que les lignes suivantes :

```
BRIDGES = {  
  
}
```

Ps : Dans cette configuration du fichier « rules.py » (vide de toutes règles), le serveur HBLink mettra en relation tous les OM's se présentant avec n'importe quel Talk Group.

**Pour la construction des règles de routage TG/TS, lire le
tutoriel « Configuration des règles d'un serveur HBLink » à
la fin de cette documentation.**



Francophone

4 - Préparation du service et lancement automatique du serveur HBLink :

```
nano /lib/systemd/system/hblink.service
```

Dans le fichier **hblink.service** qui est vide insérez, les lignes suivantes :

```
[Unit]
Description=HB bridge all Service

After=network-online.target  syslog.target
Wants=network-online.target

[Service]
StandardOutput=null
WorkingDirectory=/opt/HBlink3
RestartSec=3
ExecStart=/usr/bin/python3 /opt/HBlink3/bridge.py
Restart=on-abort

[Install]
WantedBy=multi-user.target
```

Activation du service serveur HBLink :

```
systemctl enable hblink.service
```

Lancement du service serveur HBLink :

```
systemctl start hblink.service
```

Vérification que le service fonctionne correctement :

```
systemctl status hblink.service
```

et doit répondre sans erreur de la façon suivante :

```
hblink.service - HB bridge all Service
  Loaded: loaded (/lib/systemd/system/hblink.service; enabled; vendor
  preset: enabled) Active: active (running) since Mon 2019-06-10
  13:44:38 CEST; 2h 11min ago
  Main PID: 29582 (python3)
  Tasks: 1 (limit: 4915)
         CGroup:
             /system.slice/hblink.service
                 └─29582 /usr/bin/python3 /opt/HBlink3/bridge.py
```

```
cze 10 13:44:38 dmr-torun systemd[1]: Started HB bridge all Service.
```

5 - Installation du monitor WEB du serveur HBLink.

Se placer dans le répertoire /opt

```
cd /opt
git clone https://github.com/sp2ong/HBmonitor.git
cd HBmonitor
chmod +x install.sh
./install.sh
```

Modification du fichier de configuration du monitor.

```
cp config_SAMPLE.py config.py
nano config.py
```

Vérifier les concordances des champs suivants :

```
REPORT_NAME = 'Titre de votre Dashboard'      #Name of the monitored HBLink system
CONFIG_INC = True                             # Include HBLink stats
LASTHEARD_INC = True                          # Show lastheard
HOMEBREW_INC = True                           # Include Homebrew peers
BRIDGES_INC = True                            # Include Bridge stats (confbrige.py)
HBLINK_IP = '127.0.0.1'                       # HBLink's IP Address
HBLINK_PORT= 4321                             # HBLink's TCP reporting socket
FREQUENCY = 10                               # Frequency to push updates to web clients
WEB_SERVER_PORT = 8080                       #Has to be above 1024 if you're not running as root
CLIENT_TIMEOUT = 0                           # Clients are timed out after this many seconds,0 to disable

# Files and stuff for loading alias files for mapping numbers to names

PATH = './'                                  # MUST END IN '/'
PEER_FILE = 'peer_ids.json'
SUBSCRIBER_FILE = 'subscriber_ids.json'
TGID_FILE = 'talkgroup_ids.json'
LOCAL_SUB_FILE = 'local_subscriber_ids.'
LOCAL_PEER_FILE = 'local_peer_ids.json'
FILE_RELOAD = 7
PEER_URL = 'https://www.radioid.net/static/rptrs.json'
SUBSCRIBER_URL= 'https://www.radioid.net/api/dmr/user/?country=Poland'

# Settings for log files

LOG_PATH = './log/'
LOG_NAME = 'hbmon.log'
```

6 - Préparation du service et lancement automatique du monitor HBLink :

```
nano /lib/systemd/system/hbmon.service
```

Dans le fichier **hbmon.service** qui est vide, insérer les lignes suivantes :

```
[Unit]
Description=HBmon Service

After=network-online.target syslog.target
Wants=network-online.target

[Service]
StandardOutput=null
WorkingDirectory=/opt/HBmonitor
RestartSec=3
ExecStart=/usr/bin/python3 /opt/HBmonitor/monitor.py
Restart=on-abort

[Install]
WantedBy=multi-user.target
```

Activation du service monitor HBLink :
`systemctl enable hbmon.service`

Lancement du service monitor HBLink :
`systemctl start hbmon.service`

Vérification que le service fonctionne correctement :
`systemctl status hbmon.service`
qui doit répondre **sans erreur** de la façon suivante :

```
hbmon.service - HB bridge all Service
Loaded: loaded (/lib/systemd/system/hbmon.service; enabled;
vendor preset: enabled) Active: active (running) since Mon
2019-06-10 13:42:06 CEST; 2h 18min ago
Main PID: 29543
  (python) Tasks: 1
  (limit: 4915)
      CGroup:
      /system.slice/hbmon.service
          └─29543 /usr/bin/python3
              /opt/HBmonitor/monitor.py
```

```
cze 10 13:42:06 dmr-torun systemd[1]: Started HB bridge all Service.
```

A partir de maintenant il est possible se de connecter au moniteur du serveur HBLink avec l'URL suivante :

http://adresse_IP_du_serveur_VPS:8080

ANNEXES

7 - Commandes Linux de gestion des services :

Syntaxe : `systemctl <commande> <service>`

Les deux services nécessaires, **hblink.service** et **hbmon.service**, sont gérés avec la commande **systemctl** standard :

Les commandes sont :

start	Démarre le service
stop	Arrête le service
restart	Redémarre le service
status	Montre le statut de fonctionnement
enable	Démarre le service automatiquement au démarrage
disable	Désactive le démarrage automatique

Exemple : Vous voulez redémarrer le serveur hblink et vérifier que tout soit OK.

Avec les droits root :

```
systemctl restart hblink.service
```

Puis :

```
systemctl status hblink.service
```

8 - Gestion de la rotation des logs :

En production un serveur génère beaucoup de fichiers journaux (logs) qui peuvent prendre au bout de quelques mois une taille importante sur le disque du serveur. Un serveur qui n'a plus assez de place pour travailler s'arrête purement et simplement. Il est donc indispensable d'assurer une gestion de ces logs.

En standard, un programme est fait pour cela (**logrotate**)

En prérequis, nous considérons que vous avez suivi ce tutoriel et que vous avez paramétré vos services hblink et hbmon pour que les logs de ces derniers s'enregistrent dans : `/var/log/`

Positionnez-vous dans `/etc/logrotate.d/` avec la commande : `cd /etc/logrotate.d`

Avec nano ouvrez un fichier nommé **hblink** avec la commande : `nano hblink` ,et inscrivez à l'intérieur les lignes suivantes :

```
/var/log/hblink.log
{
copytruncate
delaycompress
rotate 7
daily
compress
missingok
notifempty
}
```

Idem en créant un fichier **hbmon** avec la commande `nano hbmon` et inscrivez à l'intérieur les lignes suivantes :

```
/var/log/hbmon.log
{
copytruncate
delaycompress
rotate 7
daily
compress
missingok
notifempty
}
```

Automatiquement, tous les jours de la semaine, **logrotate** fera une copie des fichiers **hblink.org** et **hbmon.org** et les compressera. Il faudra néanmoins un jour, les effacer manuellement ou automatiquement.

<https://doc.ubuntu-fr.org/logrotate>

Un autre fichier dont la taille doit être vérifiée est **lastheard.log** qui se trouve dans le répertoire du journal de **HBmonitor**. `/opt/HBmonitor/log`

Pour ce faire, il y a un script dans le répertoire `/opt/HBmonitor/utills`, appelé **lastheard** (fichier de connexion des derniers OM's entendu).

Il est nécessaire, après avoir ajusté les chemins des logs qu'il contient, de le rendre exécutable (`chmod +x lastheard`) et de le copier dans `/etc/cron.daily/` et enfin de recharger le fichier « cron » ou en redémarrant votre serveur.

Contenu du fichier **lastheard** :

```
#!/bin/bash
mv /opt/HBmonitor/log/lastheard.log
/opt/HBmonitor/log/lastheard.log.save
/usr/bin/tail -150 /opt/HBmonitor/log/lastheard.log.save >
/opt/HBmonitor/log/lastheard.log

mv /opt/HBmonitor/log/lastheard.log
/opt/HBmonitor/log/lastheard.log.save
/usr/bin/tail -150 /opt/HBmonitor/log/lastheard.log.save >
/opt/HBmonitor/log/lastheard.log
```

9 - La personnalisation du tableau de bord du moniteur WEB

C'est dans le fichier `/opt/HBmonitor/index_template.html`

Pour le personnaliser, éditez simplement le fichier `index_template.html` dans son répertoire principal, en suivant les exemples commentés à l'intérieur.

Une fois modifié, le service **hbmon** doit être redémarré :

```
systemctl restart hbmon.service
```

puis

```
systemctl status hbmon.service
```

L'image de logo est codée en Base64 et codé en dur dans le fichier `index_template.html`

Pour transcoder un logo « jpg » en base 64 :

<https://www.base64-image.de>

et copier-coller le résultat en remplaçant l'ancien dans le fichier `index_template.html`



FONCTIONS AVANCEES

10 - Installation du service ECHO (Parrot) Service HBLink Parrot

Le service **PARROT** va permettre aux utilisateurs de votre serveur de faire des tests de modulation et de s'écouter en retour. Cela se passe sur un Talk Group particulier (Ex : TG9990/TS2) qui permet les essais sans « polluer » les autres TG alloués à du trafic ordinaire.

Il faut tout d'abord vérifier que les paquets suivants sont bien installés :

Sous utilisateur root

```
apt-get install python3-bitstring
apt-get install python3-bitarray
apt-get install python3-twisted
pip install configparser
```

-Installez le package Parrot à notre serveur :

```
cd /opt/HBlink3
wget https://raw.githubusercontent.com/n0mjs710/hblink3/master/playback.py
chmod +x playback.py
```

-Éditez le fichier **playback.cfg** : `nano opt/HBlink3/playback.cfg` et faites un « copier-coller » des lignes ci-dessous.

```
[GLOBAL]
PATH: ./
PING_TIME: 10
MAX_MISSED: 5
USE_ACL: True
REG_ACL: PERMIT:ALL
SUB_ACL: DENY:1
TGID_TS1_ACL: PERMIT:ALL
TGID_TS2_ACL: PERMIT:9990
```

```
[REPORTS]
REPORT: False
REPORT_INTERVAL: 60
REPORT_PORT: 4322
REPORT_CLIENTS: 127.0.0.1
```

```
[LOGGER]
LOG_FILE: /var/log/parrot.log
LOG_HANDLERS: file-timed
LOG_LEVEL: INFO
LOG_NAME: Parrot
```

```
[ALIASES]
TRY_DOWNLOAD: False
```

```
PATH: ./
PEER_FILE: peer_ids.json
SUBSCRIBER_FILE: subscriber_ids.json
TGID_FILE: talkgroup_ids.json
PEER_URL: https://www.radioid.net/static/rptrs.json
SUBSCRIBER_URL: https://www.radioid.net/api/dmr/user/?country=Poland
STALE_DAYS: 7
```

```
[PARROT]
MODE: MASTER
ENABLED: True
REPEAT: True
MAX_PEERS: 10
EXPORT_AMBE: False
IP:
PORT: 54100      #Port sur lequel le playback enverra le flux au serveur HBLink
PASSPHRASE: password
GROUP_HANGTIME: 5
USE_ACL: True
REG_ACL: PERMIT:9990
SUB_ACL: DENY:1
TGID_TS1_ACL: DENY:ALL
TGID_TS2_ACL: PERMIT:9990
```

-Éditez le fichier **parrot.service** `nano /lib/systemd/system/parrot.service` et faites un « copier-coller » des lignes ci-dessous :



[Unit]
Description=HB bridge all Service

After=network-online.target syslog.target
Wants=network-online.target

[Service]
StandardOutput=null
WorkingDirectory=/opt/HBlink3
RestartSec=3
ExecStart=/usr/bin/python3 /opt/HBlink3/playback.py -c
/opt/HBlink3/playback.cfg
Restart=on-abort

[Install]
WantedBy=multi-user.target

Activez le service PARROT : `systemctl enable parrot.service`
Lancez le service PARROT : `systemctl start parrot.service`
Testez le fonctionnement du service PARROT : `systemctl status parrot.service`

Si tout Ok, éditez le fichier de configuration du serveur HBLink :

`nano /opt/HBlink3/hblink.cfg` afin d'y insérer le service PARROT

```
[PARROT]
MODE: PEER
ENABLED: True
LOOSE: False
EXPORT_AMBE: False
IP:
PORT: 54098           #Port de diffusion
MASTER_IP: 127.0.0.1
MASTER_PORT: 54100  #Port sur lequel le serveur reçoit le flux du playback
PASSPHRASE: password #Mot de passe identique à celui du playback.cfg
CALLSIGN: PARROT
RADIO_ID: 9990
RX_FREQ: 000
TX_FREQ: 000
TX_POWER: 5
COLORCODE: 1
SLOTS: 2
LATITUDE: 53.00000
LONGITUDE: -8.00000
HEIGHT: 0
LOCATION: Server PARROT: TG 9990
DESCRIPTION: Parrot
URL:
SOFTWARE_ID: DMRGateway-20190620
PACKAGE_ID: MMDVM_HBlink
GROUP_HANGTIME: 5
OPTIONS:
USE_ACL: False
SUB_ACL: DENY:1
TGID_TS1_ACL: PERMIT:ALL
TGID_TS2_ACL: PERMIT:ALL
```



Maintenant, créez les règles pour votre TG 9990 en éditant le fichier `rules.py` :

`nano /opt/HBlink3/rules.py`

Nota: Le fichier ci-dessous est donné à titre d'exemple et doit être adapté à votre configuration !!

```
'PARROT': [  
{'SYSTEM': 'MASTER-HS','TS':2,'TGID':9990,'ACTIVE': False,'TIMEOUT':1,'TO_TYPE': 'NONE','ON':[9990],'OFF':[],'RESET':[]},  
{'SYSTEM': 'MASTER-RPT','TS':2,'TGID':9990,'ACTIVE':False,'TIMEOUT':1,'TO_TYPE':'NONE','ON':[9990],'OFF':[],'RESET':[]},  
{'SYSTEM': 'PARROT','TS':2,'TGID':9990,'ACTIVE':True,'TIMEOUT':1,'TO_TYPE':'NONE','ON':[],'OFF':[] 'RESET':[]},  
],
```

-Relancez votre serveur HBLink: `systemctl restart hblink.service`

-Testez votre serveur HBLink: `systemctl status hblink.service`

Si tout est OK rebootez votre serveur : `/usr/sbin/init 6`

-Ouvrez votre Moniteur WEB et vérifiez que votre service PARROT (Id : 9990) est bien affiché dans la zone **HB Protocol Peer Systems**



11 - Interconnexion de 2 serveurs HBLink avec OpenBridge

Section OpenBridge votre serveur :

Éditez le fichier hblink.cfg puis ajoutez l'entrée suivante :

```
[OPB1]
MODE: OPENBRIDGE
ENABLED: True
IP:
PORT: 62100 #Port sur lequel le serveur distant se connecte à votre serveur
NETWORK_ID: 208xxxxx01 #Votre Net ID
PASSPHRASE: #Mot de passe commun au 2 serveurs
TARGET_IP: #Adresse IP du serveur distant
TARGET_PORT: 62150 #Port de connexion sur le serveur distant
USE_ACL: False
SUB_ACL: DENY:1
TGID_ACL: PERMIT:ALL
```

Section OpenBridge sur le serveur distant :

L'administrateur du serveur distant doit faire la même opération sur son serveur en adaptant l'IP et les ports à vos paramètres.

```
[OPB1]
MODE: OPENBRIDGE
ENABLED: True
IP:
PORT: 62150 #Port sur lequel votre serveur se connecte
NETWORK_ID: #Le Network ID de ce serveur
PASSPHRASE: #Mot de passe commun au 2 serveurs
TARGET_IP: #Adresse IP de votre serveur
TARGET_PORT: 62100 #Port de connexion sur votre serveur
USE_ACL: False
SUB_ACL: DENY:1
TGID_ACL: PERMIT:ALL
```

- Éditez les règles dans le fichier rules.py : `nano /opt/HBlink3/rules.py`

Nota: Le fichier ci-dessous est donné à titre d'exemple et doit être adapté à votre configuration !!

```
'TG7': [
{'SYSTEM': 'MASTER-HS', 'TS': 2, 'TGID': 7, 'ACTIVE': True, 'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': []},
{'SYSTEM': 'MASTER-RPT', 'TS': 2, 'TGID': 7, 'ACTIVE': True, 'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': []},
{'SYSTEM': 'OPB1', 'TS': 1, 'TGID': 7, 'ACTIVE': True, 'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': [], } ]
```

Attention les systèmes raccordés sur OPENBRIDGE ne communiquent que sur TS1

-Relancez votre serveur HBLink: `systemctl restart hblink.service`

-Testez votre serveur HBLink: `systemctl status hblink.service`

Si tout est OK rebootez votre serveur : `/usr/sbin/init 6`

-Ouvrez votre Moniteur WEB et vérifiez que **OBP1** est bien affiché dans la zone **OpenBridge Systems**.

12 - Interconnexion avec un serveur Brandmeister

Section OpenBridge votre serveur :

-Editez le fichier hblink.cfg puis ajoutez l'entrée suivante :

```
[OPB-BM]
MODE: OPENBRIDGE
ENABLED: True
IP:
PORT: 62035 #Votre port d'écoute
NETWORK_ID: 208xxxx50 #Votre Net Id donné par le serveur distant
PASSPHRASE: * #Password à demander au serveur distant
TARGET_IP: #Adresse IP du serveur distant
TARGET_PORT: 62031 #Port de connexion sur le serveur distant
USE_ACL: False
SUB_ACL: DENY:1
TGID_ACL: PERMIT:ALL
```

- Éditez vos règles

Attention les systèmes raccordés sur OPENBRIDGE ne communiquent que sur TS1

-Relancez votre serveur HBLink: `systemctl restart hblink.service`

-Testez votre serveur HBLink: `systemctl status hblink.service`

Si tout est OK rebootez votre serveur : `/usr/sbin/init 6`

-Ouvrez votre Moniteur WEB et vérifiez que **OBP-BM** est bien affiché dans la zone « **OpenBridge Systems** »

13 - Relais ou Hotspot duplex connecté sur votre serveur HBLink et sur un serveur distant afin d'en récupérer les flux d'un TG (sans OpenBridge).

ATTENTION cette utilisation est proscrite par Brandmeister et donnée ici uniquement dans un but d'exemple

-Programmez votre Hotspot duplex (hotspot physique tel que pi-star HS Hat Dual) pour qu'il se connecte sur votre serveur HBLink sur un port dédié ex : 62030

```
[F1ZXX]
MODE: MASTER
ENABLED: True
REPEAT: True
MAX_PEERS: 3
EXPORT_AMBE: False
IP:
PORT: 62030
PASSPHRASE: passw0rd
GROUP_HANGTIME: 1
USE_ACL: True
REG_ACL: PERMIT:208xxxx
SUB_ACL: DENY:1
TGID_TS1_ACL: PERMIT:ALL
TGID_TS2_ACL: PERMIT:ALL
```

DMR
Francophone

-Créez un Hotspot vituel dans votre hblink.cfg qui lui, va se connecter sur le serveur Brandmeister ou HBLink distant avec le même indicatif (CALLSIGN)

```
[PEER-BM208]
MODE: PEER
ENABLED: True
LOOSE: False
EXPORT_AMBE: False
IP:
PORT: 54003 #Port dédié sur le serveur distant à défaut 62031
MASTER_IP: #Adresse IP du serveur HBLink distant
MASTER_PORT: 62030
PASSPHRASE: passw0rd
CALLSIGN: F1ZXX
RADIO_ID: 208xxx02 #Votre DMR ID avec un suffixe 02,03, 04...99
RX_FREQ: 430500000
TX_FREQ: 439900000
TX_POWER: 10
COLORCODE: 1
SLOTS: 1
LATITUDE: 45.8215
LONGITUDE: 004.8660 # ATTENTION pour un raccordement sur BM ajouter 00
HEIGHT: 1
LOCATION: Lyon
DESCRIPTION: Cool repeater
URL:
SOFTWARE_ID: 20170620
PACKAGE_ID: MMDVM_HBlink
GROUP_HANGTIME: 5
OPTIONS:
USE_ACL: True
SUB_ACL: DENY:1
TGID_TS1_ACL: PERMIT:ALL
TGID_TS2_ACL: PERMIT:ALL
```



-Éditez vos règles

-Relancez votre serveur HBLink: `systemctl restart hblink.service`

-Testez votre serveur HBLink: `systemctl status hblink.service`

Si tout est OK rebootez votre serveur : `/usr/sbin/init 6`

-Ouvrez votre Moniteur WEB

Vérifiez que **PEER-BM208** est bien affiché dans la zone « **HB Protocol Peer Systems** »

Et que

F1ZXX (Id 208xxx02) s'affiche dans la zone « **Callsign (DMR Id : 208xxx02)** »

Paramètres de configurations d'un serveur HBLink3

Configuration du serveur : hblink.cfg

Le fichier de configuration du serveur se trouve dans [/opt/Hblink3/hblink.cfg](#)

Section [GLOBAL]

[GLOBAL]

PATH: ./

PING_TIME: 5

MAX_MISSED: 3

USE_ACL: True

REG_ACL: PERMIT: ALL

SUB_ACL: DENY: 1

TGID_TS1_ACL: PERMIT: ALL

TGID_TS2_ACL: PERMIT: ALL

PATH - Chemin de travail pour les fichiers. Inutile de le changer.

PING_TIME - Intervalle de temps pendant lequel les « peers connectés » vont envoyer une requête ping au maître et réessayer l'enregistrement (5 essais). Inutile de changer.

MAX_MISSED – Fixe le nombre de tentatives connexions manquées d'un « peer », avant de le déconnecter.

Les ACL's sont des listes de contrôle d'accès et sont un outil très puissant pour administrer votre système. Attention de ne pas en abuser car elles consomment du traitement CPU. Si des ACL ne sont plus utiles, désactivez-les

```
# L'action peut être PERMIT ou DENY
# Chaque entrée peut être un identifiant radio unique ou une plage
  (par exemple 1-2999)
# Format des ACL = 'action: id | début-fin |, id | début-fin, ....'
# SUB_ACL: DENY: 1,1000-2000,4500-60000,17
```

Les différents types d'ACL :

REG_ACL: ID radio homologues pour l'enregistrement (utilisé uniquement sur les systèmes maîtres HBP)

SUB_ACL: ID d'abonné pour les utilisateurs finaux

TGID_TS1_ACL: ID de Talk Group sur TimeSlot1

TGID_TS2_ACL: ID de Talk Group sur TimeSlot2

Les **ACL globales** seront toujours traitées **AVANT les ACL de niveau (hotspots, relais..)**

Les paquets sont tout d'abord comparés aux ACL Globales puis aux suivantes.

Si un paquet « *passe* » l'ensemble des ACL, le traitement se poursuit, dans le cas contraire il est rejeté.

Si vous ne souhaitez pas utiliser les ACL, réglez-les sur 'PERMIT: ALL'

TGID_TS1_ACL dans la section [GLOBAL] est utilisée pour les systèmes **OPENBRIDGE**, car tout le trafic est transmis uniquement le TS1 entre système « OpenBridge »

La section [GLOBALE] reste en générale non modifiée.

Pour plus de granularité, les ACL d'accès peuvent, être renseignées dans les sections individuelles (hotspots, relais..) comme dans l'exemple suivant :

Exemple : configuration de la zone HOTSPOT-Simplex.

```
[HOTSPOT-Simplex]                                # Zone Système
MODE: MASTER                                       # Connexion sur le serveur HBLink Master
ENABLED: True ou False                             # Activation ou désactivation de cette zone
REPEAT: True ou False                             # Le trafic est répété sur tous les hotspots /ports connectés à cette zone ou isolé (False)
MAX_PEERS: 30                                      # Nombre de hotspots maximum dans cette zone
EXPORT_AMBE: False                                # Toujours laisser à « False »
IP:                                                 # Adresse de connexion IP interne si NAT (sinon laisser vide)
PORT: 62031                                        # Port de connexion
PASSPHRASE: passw0rd                               # Mot de passe de connexion
GROUP_HANGTIME: 1                                 # ???
USE_ACL: True                                       # On active les ACL pour cette zone
REG_ACL: DENY:1,208000xxx,2087xxx                 # Ici 2 ID sont interdit de connexion (208000xxx,2087xxx)
SUB_ACL: DENY:1                                    # ???
TGID_TS1_ACL: DENY:ALL                             # Hotspot simplex TS1 inutile
TGID_TS2_ACL: PERMIT:900,4000                     # Les Hotspots-simplex ne peuvent communiquer que TS2
```

Nota : Tous les hotspots connectés à une même zone partagent les mêmes flux

Pour éviter cela, il est possible de créer des zones particulières (sur des ports différents).

Exemple : configuration d'une zone [RELAIS]

```
[R-F1ZXX]
MODE: MASTER
ENABLED: True
REPEAT: True
MAX_PEERS: 1
EXPORT_AMBE: False
IP:
PORT: 62030                                        #Un port unique pour cette zone
PASSPHRASE: passw0rd
GROUP_HANGTIME: 1
USE_ACL: True                                       #Activation de ACL ci-dessous
REG_ACL: DENY:1
SUB_ACL: DENY:1
TGID_TS1_ACL: PERMIT:901-909,4000                 #Sur TS1 permission des TG901 à 909 et 4000 (déconnexion)
TGID_TS2_ACL: PERMIT:900,4000                     #Sur TS2 permission du TG900 uniquement et déconnexion
```

Exemple : configuration de la zone [PEER] (relais virtual) # Cette zone sera connecté à un autre serveur

```
[PEER-1]
MODE: PEER
ENABLED: False           #Activation ou désactivation de cette zone
LOOSE: False            # A laisser "False" sauf en cas de connexion sur un réseau non compatible 100%
EXPORT_AMBE: False     # A laisser "False"
IP:                     # A laisser vide
PORT: 54001             # port unique à fixer sur le serveur HBLink
MASTER_IP: 80.12.XX.XX #Adresse IP du serveur auquel nous voulons nous connecter
MASTER_PORT: 54000     #Port du serveur auquel nous voulons nous connecter
PASSPHRASE: Toto       #Le mot de passe du serveur auquel nous voulons nous connecter
CALLSIGN: F1XXX        #Indicatif avec lequel nous voulons nous connecter (les suffixes sont acceptés)
RADIO_ID: 208XXXX      #DMR Id avec lequel nous voulons nous connecter
RX_FREQ: 439000000
TX_FREQ: 430000000
TX_POWER: 25
COLORCODE: 1
SLOTS: 1                # 1=TS1    2=TS2    3=TS1+TS2
LATITUDE: 38.0000
LONGITUDE: -095.0000
HEIGHT: 75
LOCATION: Lyon, FR
DESCRIPTION: Relais virtuel
URL: dmr-francophone.net
SOFTWARE_ID: 20170620
PACKAGE_ID: MMDVM_HBlink
GROUP_HANGTIME: 5
OPTIONS:
USE_ACL: True
SUB_ACL: DENY:1
TGID_TS1_ACL: PERMIT:ALL
TGID_TS2_ACL: PERMIT:ALL
```



DMR
Francophone

Configuration des règles : rules.py

Le fichier de configuration des règles de routage des flux se trouve </opt/Hblink3/rules.py>

SYSTEM - Le nom du système tel qu'il est répertorié dans le fichier de configuration principal de hblink.cfg)

Il DOIT être exactement du même nom que dans le fichier de configuration principal !!!

TS - Time slot utilisé pour faire correspondre le trafic à ce pont de conférence. (Les connexions XLX doivent TOUJOURS utiliser uniquement le TS 2).

TGID – Numéro de Talk Group utilisé (TG).

(Les connexions XLX doivent TOUJOURS et uniquement utiliser le TG 9).

ON et **OFF** sont utilisés pour allumer ou éteindre un TG Même si vous n'en voulez qu'un, il doit être au format liste. []

TO_TYPE (ON/OFF/NONE) est de type timeout. Si vous souhaitez utiliser des minuteriers, ON signifie que cette fonction est activée. Le TG sera éteinte après le délai d'attente. OFF signifie qu'il se rallumera après le délai d'attente. Si vous ne voulez pas utiliser de minuteriers, réglez-le sur autre chose, mais 'NONE' peut être une bonne valeur pour la documentation !

TIMEOUT est une valeur en minutes pour le temporisateur.

Attention, l'usage des minuteriers est gourmande en ressources CPU et donc peut avoir une action sur la performance du serveur HBLink.

RESET est une liste d'ID de groupe de conversation qui, en plus des listes ON et OFF, provoquera une minuterie à réinitialiser. Ceci est utile si vous utilisez des TGID différents pour le déclenchement du trafic vocal.

Si vous n'en avez pas besoin, n'utilisez pas cette fonction.

Explication des règles par l'exemple

TG222: [
{'SYSTEM': ' M-SIMPLEX ', 'TS': 2, 'TGID': 222, 'ACTIVE': False, 'TIMEOUT': 10, 'TO_TYPE': 'ON', 'ON': [222,], 'OFF': [4000,], 'RESET': []},

'TG222': [

*C'est le nom du flux : Vous pouvez mettre n'importe quoi comme nom mais il **doit être unique**.*

{'SYSTEM': ' M-SIMPLEX ', 'TS': 2, 'TGID': 222, 'ACTIVE': False,

Les flux arrivant sur la zone M-SIMPLEX doivent être sur TS2 et après avoir défini le TG222 ('TGID':222). Ce TG ne doit pas être actif au démarrage ('ACTIVE':False). Il sera activé dynamiquement avec un coup de PTT

'TIMEOUT': 10, 'TO_TYPE': 'ON', 'ON': [222,], 'OFF': [4000,], 'RESET': []},

Au bout de 10 minutes (TIMEOUT: 10), les TG dynamiques seront éteints (TO_TYPE:'ON'). Le TG222 pourra être réactivé avec un coup de PTT ('ON':[222]). Les TG pourront être forcés à la déconnexion avec le TG4000 ('OFF':[4000,]) avant l'expiration du délai.

{'SYSTEM': ' M-DUPLEX ', 'TS': 1, 'TGID': 222, 'ACTIVE': True, 'TIMEOUT': 10, 'TO_TYPE': 'NONE', 'ON': [], 'OFF': [], 'RESET': []},

Tous les flux arrivant sur la zone M-DUPLEX, seront sur TS1 / TG222 statique au démarrage, le timeout ne sera pas pris en compte ('TO_TYPE':'NONE'), jamais éteint ('ON': []) et sans possibilité de le déconnecter ('OFF':[]).

{'SYSTEM': ' BM-2222-01 ', 'TS': 1, 'TGID': 222, 'ACTIF': True, 'TIMEOUT': 10, 'TO_TYPE':'NONE', 'ON': [], 'OFF': [], 'RESET': []},

Le flux acquis par BM ('BM-2222-01') sera transmis sur TS1/ TG222, statique et monté au démarrage ('ACTIF':True).Le timeout ne sera pas pris en compte et le TG222 ne sera jamais éteint et sans possibilité de le déconnecter (La connexion externe vers BM sera toujours active)

31 janvier 2020 - F1FQN – Patrick / pour les réseaux :

[HBLink DMR Francophone](#)

&

[IPSC2-FR-LINK](#)