

NEmu

NEmu : un outil de virtualisation de réseaux à la demande pour l'enseignement

Vincent Autefage
autefage@labri.fr

Damien Magoni
magoni@labri.fr

<http://nemu.valab.net>



De quoi n'allons-nous pas parler ?



Virtualisation système (VirtualBox, OpenVZ, XEN)



Cloud commercial (Amazon, Google, Microsoft, OVH)



Gestionnaire IAAS (OpenNebula, OpenStack, VMWare)



Plateforme collaborative (Emulab, GENI, PlanetLab)



Émulation réseau matérielle (Apposite, PacketStorm, OpenFlow)

De quoi n'allons-nous pas parler ?



Virtualisation système (VirtualBox, OpenVZ, XEN)



Cloud commercial (Amazon, Google, Microsoft, OVH)



Gestionnaire IAAS (OpenNebula, OpenStack, VMWare)



Plateforme collaborative (Emulab, GENI, PlanetLab)



Émulation réseau matérielle (Apposite, PacketStorm, OpenFlow)

De quoi n'allons-nous pas parler ?



Virtualisation système (VirtualBox, OpenVZ, XEN)



Cloud commercial (Amazon, Google, Microsoft, OVH)



Gestionnaire IAAS (OpenNebula, OpenStack, VMWare)



Plateforme collaborative (Emulab, GENI, PlanetLab)



Émulation réseau matérielle (Apposite, PacketStorm, OpenFlow)

De quoi n'allons-nous pas parler ?



Virtualisation système (VirtualBox, OpenVZ, XEN)



Cloud commercial (Amazon, Google, Microsoft, OVH)



Gestionnaire IAAS (OpenNebula, OpenStack, VMWare)



Plateforme collaborative (Emulab, GENI, PlanetLab)



Émulation réseau matérielle (Apposite, PacketStorm, OpenFlow)

De quoi n'allons-nous pas parler ?



Virtualisation système (VirtualBox, OpenVZ, XEN)



Cloud commercial (Amazon, Google, Microsoft, OVH)



Gestionnaire IAAS (OpenNebula, OpenStack, VMWare)



Plateforme collaborative (Emulab, GENI, PlanetLab)



Émulation réseau matérielle (Apposite, PacketStorm, OpenFlow)

De quoi n'allons-nous pas parler ?



Virtualisation système (VirtualBox, OpenVZ, XEN)



Cloud commercial (Amazon, Google, Microsoft, OVH)



Gestionnaire IAAS (OpenNebula, OpenStack, VMWare)



Plateforme collaborative (Emulab, GENI, PlanetLab)



Émulation réseau matérielle (Apposite, PacketStorm, OpenFlow)



Contexte

Apprentissage et pratique

- **lien étroit** établi depuis de nombreuses années¹ ;
- **pratique active** augmenterait de **84%** les **résultats** scolaires² ;
- d'autant plus vrai pour l'**informatique**³.

Mise en œuvre difficile

- droits administrateurs ;
- mobilité ;
- propriétés systèmes et réseaux ;
- etc.

1. Gérard Barnier. *Philosophie de l'éducation grands courants pédagogiques*. IUFM Aix Marseille, 2003.

2. Jean-Michel Fourgous. *Apprendre autrement à l'ère numérique*. Rapport de la mission parlementaire Fourgous, 2012.

3. Orit Hazzan, Tami Lapidot, et Noa Ragonis. *Guide to Teaching Computer Science, An Activity-Based Approach*. Springer, 2011.

Contexte

Apprentissage et pratique

- **lien étroit** établi depuis de nombreuses années¹ ;
- **pratique active** augmenterait de **84%** les **résultats** scolaires² ;
- d'autant plus vrai pour l'**informatique**³.

Mise en œuvre difficile

- droits administrateurs ;
- mobilité ;
- propriétés systèmes et réseaux ;
- etc.

1. Gérard Barnier. *Philosophie de l'éducation grands courants pédagogiques*. IUFM Aix Marseille, 2003.

2. Jean-Michel Fourgous. *Apprendre autrement à l'ère numérique*. Rapport de la mission parlementaire Fourgous, 2012.

3. Orit Hazzan, Tami Lapidot, et Noa Ragonis. *Guide to Teaching Computer Science, An Activity-Based Approach*. Springer, 2011.

Contexte

Apprentissage et pratique

- **lien étroit** établi depuis de nombreuses années¹ ;
- **pratique active** augmenterait de **84%** les **résultats** scolaires² ;
- d'autant plus vrai pour l'**informatique**³.

Mise en œuvre difficile

- droits administrateurs ;
- mobilité ;
- propriétés systèmes et réseaux ;
- etc.

⇒ **Charges pour l'administrateur**



1. Gérard Barnier. *Philosophie de l'éducation grands courants pédagogiques*. IUFM Aix Marseille, 2003.
2. Jean-Michel Fourgous. *Apprendre autrement à l'ère numérique*. Rapport de la mission parlementaire Fourgous, 2012.
3. Orit Hazzan, Tami Lapidot, et Noa Ragonis. *Guide to Teaching Computer Science, An Activity-Based Approach*. Springer, 2011.

Contexte

Infrastructure privée

- coût ;
- espace ;
- flexibilité très limitée.

Virtualisation privative

- flexibilité accrue ;
- haute disponibilité ;
- coût réduit ;
- etc.

Infrastructure partagée

- contrainte de disponibilité ;
- temps limité ;
- paramètres peu flexibles.

Contexte

Infrastructure privée

- coût ;
- espace ;
- flexibilité très limitée.

Infrastructure partagée

- contrainte de disponibilité ;
- temps limité ;
- paramètres peu flexibles.

Virtualisation privative

- flexibilité accrue ;
- haute disponibilité ;
- coût réduit ;
- etc.

Contexte

Infrastructure privée

- coût ;
- espace ;
- flexibilité très limitée.

Infrastructure partagée

- contrainte de disponibilité ;
- temps limité ;
- paramètres peu flexibles.

Virtualisation privative

- flexibilité accrue ;
- haute disponibilité ;
- coût réduit ;
- etc.

Contexte

Infrastructure privée

- coût ;
- espace ;
- flexibilité très limitée.

Infrastructure partagée

- contrainte de disponibilité ;
- temps limité ;
- paramètres peu flexibles.

Virtualisation privative

- flexibilité accrue ;
- haute disponibilité ;
- coût réduit ;
- etc.

⇒ **Administrateur moins sollicité**



Network Virtualization Environment (NVE)

Définition du concept

Un **NVE** héberge plusieurs **réseaux virtuels (VN)** qui sont indépendants et qui n'ont pas conscience de l'existence des autres.

Un **NVE** est défini par plusieurs **caractéristiques**.

Network Virtualization Environment (NVE)

Définition du concept

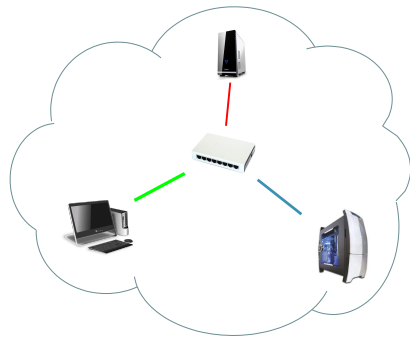
Un **NVE** héberge plusieurs **réseaux virtuels (VN)** qui sont indépendants et qui n'ont pas conscience de l'existence des autres.

Un **NVE** est défini par plusieurs **caractéristiques**.

Network Virtualization Environment (NVE)

Caractéristiques d'un NVE

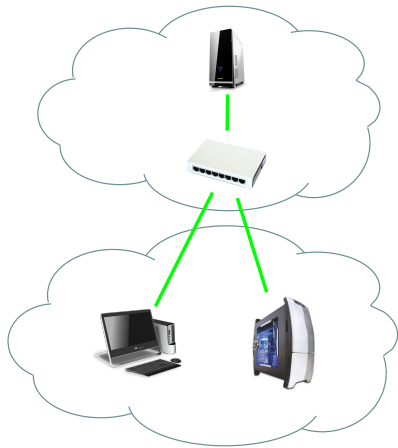
- Flexibilité & hétérogénéité
- Extensibilité
- Isolation
- Stabilité
- Maintenabilité
- Support hérité
- Programmabilité



Network Virtualization Environment (NVE)

Caractéristiques d'un NVE

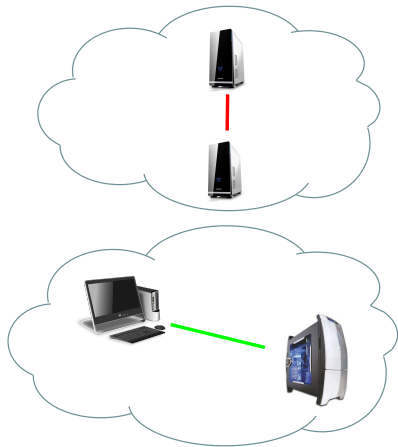
- Flexibilité & hétérogénéité
- Extensibilité
- Isolation
- Stabilité
- Maintenabilité
- Support hérité
- Programmabilité



Network Virtualization Environment (NVE)

Caractéristiques d'un NVE

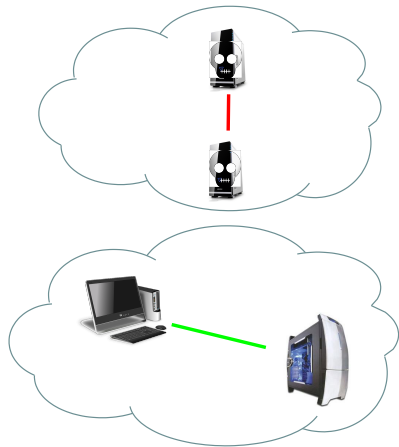
- Flexibilité & hétérogénéité
- Extensibilité
- Isolation
- Stabilité
- Maintenabilité
- Support hérité
- Programmabilité



Network Virtualization Environment (NVE)

Caractéristiques d'un NVE

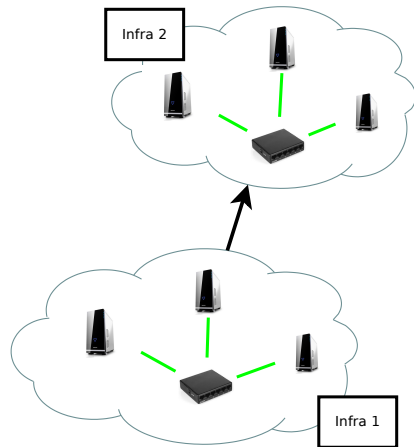
- Flexibilité & hétérogénéité
- Extensibilité
- Isolation
- Stabilité
- Maintenabilité
- Support hérité
- Programmabilité



Network Virtualization Environment (NVE)

Caractéristiques d'un NVE

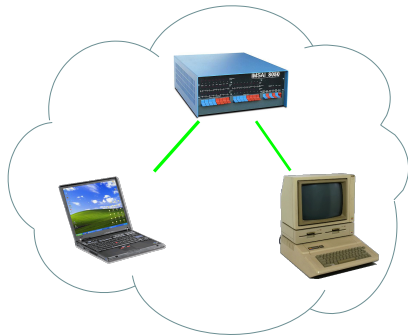
- Flexibilité & hétérogénéité
- Extensibilité
- Isolation
- Stabilité
- Maintenabilité
- Support hérité
- Programmabilité



Network Virtualization Environment (NVE)

Caractéristiques d'un NVE

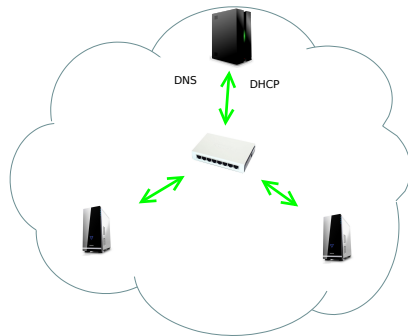
- Flexibilité & hétérogénéité
- Extensibilité
- Isolation
- Stabilité
- Maintenabilité
- Support hérité
- Programmabilité



Network Virtualization Environment (NVE)

Caractéristiques d'un NVE

- Flexibilité & hétérogénéité
- Extensibilité
- Isolation
- Stabilité
- Maintenabilité
- Support hérité
- Programmabilité



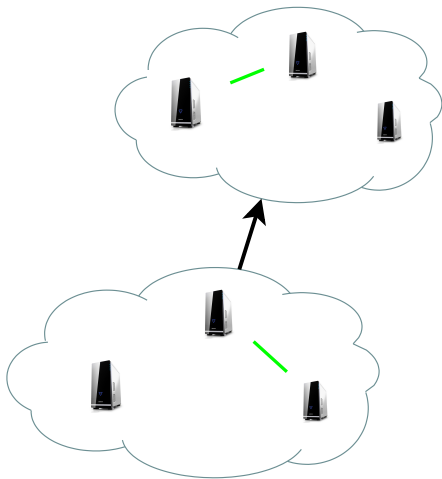
Network Virtualization Environment (NVE) +



Caractéristiques additionnelles

- Accessibilité
- Dynamicité
- Aspect communautaire
- Mobilité

Network Virtualization Environment (NVE) +



Caractéristiques additionnelles

- Accessibilité
- Dynamicité
- Aspect communautaire
- Mobilité

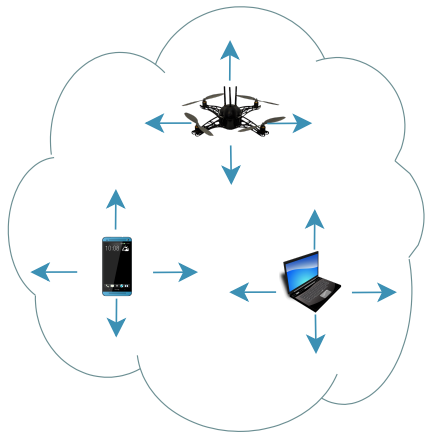
Network Virtualization Environment (NVE) +



Caractéristiques additionnelles

- Accessibilité
- Dynamicité
- Aspect communautaire
- Mobilité

Network Virtualization Environment (NVE) +



Caractéristiques additionnelles

- Accessibilité
- Dynamicit 
- Aspect communautaire
- Mobilit 

Network Virtualization Environment (NVE) +

Caractéristiques d'un NVE

- Flexibilité & hétérogénéité
- Extensibilité
- Isolation
- Stabilité
- Maintenabilité
- Support hérité
- Programmabilité

Caractéristiques additionnelles

- Accessibilité
- Dynamicité
- Aspect communautaire
- Mobilité

NEmu

<http://nemu.valab.net>

Définition

NEmu (***N**etwork **E**mulator for **m**obile **u**niverses*) est une implémentation d'un **NVE+**.

Architecture de NEmu

Nœud virtuel

Lien virtuel standard

Manager

Session

Lien virtuel mobile

Mobilisateur

Architecture de NEmu - Nœud virtuel

Définition

Un **nœud virtuel** (**VNode**) représente une **machine virtuelle** dans le réseau virtuel généré.

Implémentation

$QEMU \left\{ \begin{array}{ll} \mathbf{VHost} & \text{machine utilisateur} \\ \mathbf{VRouter} & \text{machine auto-configurée} \end{array} \right.$

Architecture de NEmu - Lien virtuel standard

Définition

Un **lien virtuel standard (VLink)** représente une **inter-connexion** entre **plusieurs entités** du réseau généré.

Implémentation

vnd	{	VLine	<i>câble Ethernet</i>
		VHub	<i>concentrateur Ethernet</i>
		VSwitch	<i>commutateur Ethernet</i>
vde	{	VDEHub	<i>concentrateur Ethernet</i>
		VDESwitch	<i>commutateur Ethernet</i>
{	VTap	<i>interface virtuelle vers le réseau réel</i>	
	VSlirp	<i>passerelle virtuelle vers le réseau réel</i>	
	VRemote	<i>pont virtuel vers une entité distante</i>	

Architecture de NEmu - Session

Définition

Une **session** représente la configuration d'une **topologie réseau** résidant sur un **même hôte physique** et appartenant à un **unique utilisateur**.

Implémentation

- 1 manager \Leftrightarrow n sessions
- Indépendante

Architecture de NEmu - Manager

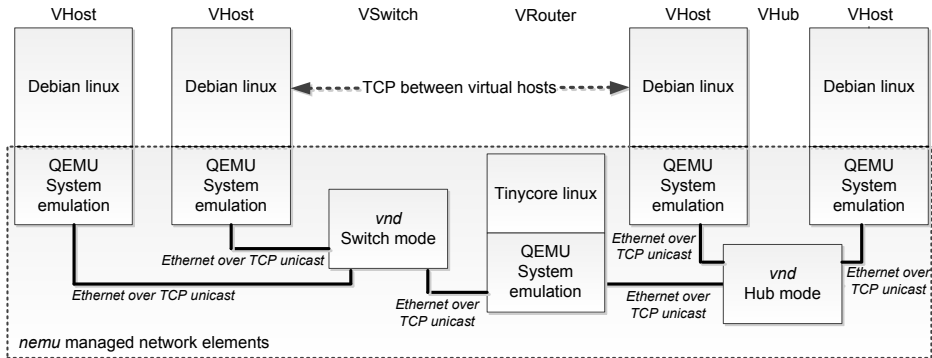
Définition

Le **manager** est le moyen de **manipuler** une session.

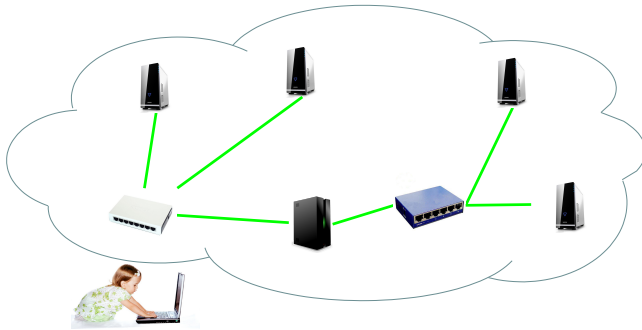
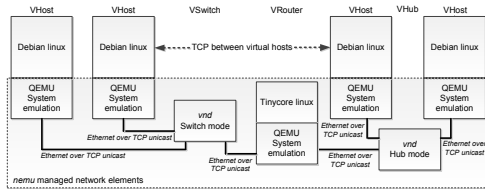
Implémentation

Python { **Module standard**
Interpreteur dynamique
Lanceur de script

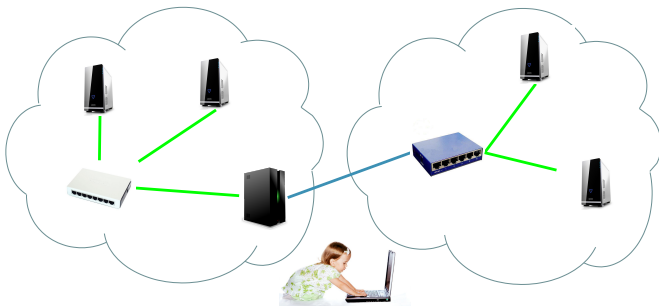
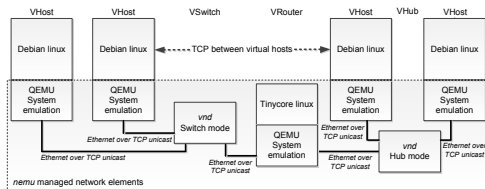
Exemple de topologie générée par NEmu



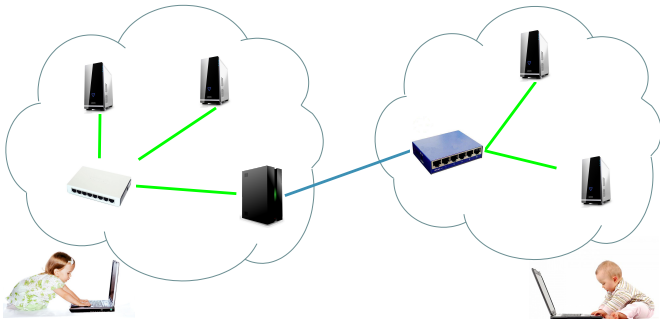
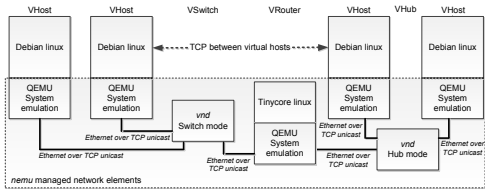
Exemple de topologie générée par NEmu



Exemple de topologie générée par NEmu



Exemple de topologie générée par NEmu



QEMU (Quick EMUlator)



- **émulateur** initié en 2003 ;
- créé par Fabrice Bellard (**France**) ;
- logiciel sous **GPL/LGPL** ;
- accéléré par **KVM** ;
- hautement **configurable** ;
- supporte un **grand nombre d'architectures** : x86-64, ARM, PPC, SPARC, MIPS, etc.
- porté sur **plusieurs OS** : Windows, Linux, Mac OS X, FreeBSD, NetBSD, OpenBSD ;
- **utilisé partiellement** au sein de VirtualBox, XEN et l'émulateur Android officiel.

VND (Virtual Network Device)

- initié pour **NEmu** en 2012 ;
- anciennement appelé **vswitch** ;
- logiciel sous **LGPL** ;
- supporte un **grand nombre de backends** : TCP, UDP, UNIX, VDE, TAP ;
- porté sur **plusieurs OS** : Windows, Linux, Mac OS X ;
- branchement **à chaud** ;
- génère des **statistiques réseaux** ;

- peut avoir des **rôles variés** :
 - câble,
 - hub,
 - switch,
 - carte sans fil,
 - point d'accès sans fil.
- réglage **unidirectionnel** :
 - bande passante,
 - délai,
 - taux d'erreur,
 - gigue,
 - taille des files,
 - VLAN de niveau 1.

VND (Virtual Network Device)

- initié pour **NEmu** en 2012 ;
- anciennement appelé **vswitch** ;
- logiciel sous **LGPL** ;
- supporte un **grand nombre de backends** : TCP, UDP, UNIX, VDE, TAP ;
- porté sur **plusieurs OS** : Windows, Linux, Mac OS X ;
- branchement **à chaud** ;
- génère des **statistiques réseaux** ;

- peut avoir des **rôles variés** :
 - câble,
 - hub,
 - switch,
 - carte sans fil,
 - point d'accès sans fil.
- réglage **unidirectionnel** :
 - bande passante,
 - délai,
 - taux d'erreur,
 - gigue,
 - taille des files,
 - VLAN de niveau 1.

Autres liens

VDE¹

- **câble**, **hub** et **switch** virtuel ;
- backend reposant sur de la **mémoire partagée**.

TAP

- interface système virtuelle **point à point** ;
- transmet des paquets entre le **noyau** et l'espace utilisateur ;
- nécessite des **droits administrateurs**.

Slirp²

- émulateur virtuel **point à point** ;
- permet une sortie vers le **réseau réel** ;
- repose sur une technique de type **NAT** ;
- émule un **relais DNS**.

Remote

- pont **point à point** imaginaire ;
- permet la **distribution** d'un réseau virtuel.

-
1. <http://vde.sourceforge.net>
 2. <http://slirp.sourceforge.net>

Autres liens

VDE¹

- **câble**, **hub** et **switch** virtuel ;
- backend reposant sur de la **mémoire partagée**.

TAP

- **interface système** virtuelle **point à point** ;
- transmet des paquets entre le **noyau** et l'espace utilisateur ;
- nécessite des **droits administrateurs**.

Slirp²

- **émulateur** virtuel **point à point** ;
- permet une sortie vers le **réseau réel** ;
- repose sur une technique de type **NAT** ;
- émule un **relais DNS**.

Remote

- pont **point à point** imaginaire ;
- permet la **distribution** d'un réseau virtuel.

-
1. <http://vde.sourceforge.net>
 2. <http://slirp.sourceforge.net>

Autres liens

VDE¹

- **câble**, **hub** et **switch** virtuel ;
- backend reposant sur de la **mémoire partagée**.

TAP

- **interface système** virtuelle **point à point** ;
- transmet des paquets entre le **noyau** et l'espace utilisateur ;
- nécessite des **droits administrateurs**.

Slirp²

- **émulateur** virtuel **point à point** ;
- permet une sortie vers le **réseau réel** ;
- repose sur une technique de type **NAT** ;
- émule un **relais DNS**.

Remote

- pont **point à point** imaginaire ;
- permet la **distribution** d'un réseau virtuel.

1. <http://vde.sourceforge.net>
2. <http://slirp.sourceforge.net>

Autres liens

VDE¹

- **câble**, **hub** et **switch** virtuel ;
- backend reposant sur de la **mémoire partagée**.

TAP

- **interface système** virtuelle **point à point** ;
- transmet des paquets entre le **noyau** et l'espace utilisateur ;
- nécessite des **droits administrateurs**.

Slirp²

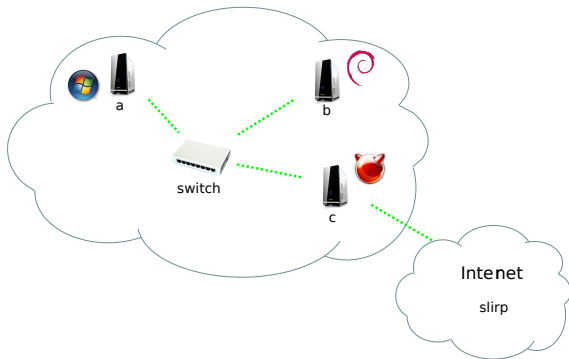
- **émulateur** virtuel **point à point** ;
- permet une sortie vers le **réseau réel** ;
- repose sur une technique de type **NAT** ;
- émule un **relais DNS**.

Remote

- pont **point à point** imaginaire ;
- permet la **distribution** d'un réseau virtuel.

1. <http://vde.sourceforge.net>
2. <http://slirp.sourceforge.net>

NEmu en pratique - Scénario filaire simple



```
InitNemu()
```

```
VSwitch('switch', niface=3)  
VSlirp('slirp')
```

```
VHost('a', hds=[VFs('windows.img', type='cow')]  
        , nics=[VNic()]  
        , m=2048, cpu='Nehalem')
```

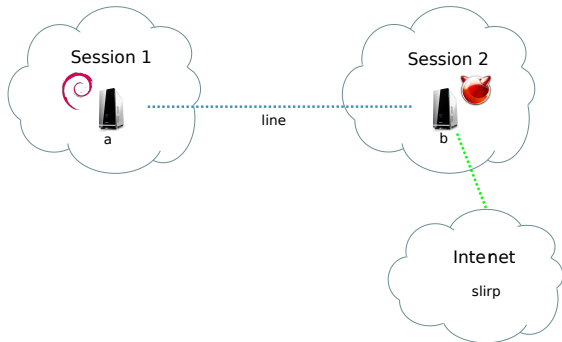
```
VHost('b', hds=[VFs('debian.img', type='cow')]  
        , nics=[VNic()]  
        , m=512, cpu='n270')
```

```
VHost('c', hds=[VFs('freebsd.img', type='cow')]  
        , nics=[VNic(), VNic()]  
        , m=1024, cpu='SandyBridge')
```

```
Link('a', 'switch:0')  
Link('b', 'switch:1')  
Link('c:0', 'switch:3')  
Link('c:1', 'slirp')
```

```
StartNemu()
```

NEmu en pratique - Scénario filaire distribué



```
#Session 1 [192.168.0.1]

InitNemu()
VLine('line')
VHost('a', hds=[VFs('debian.img', type='cow')]
      , nics=[VNic(hw='a0:00:00:00:00:01')]
      , m=512, cpu='n270')

Link('a', 'line:0')

SetIface('line:1', addr='192.168.0.2', port=7000)

StartNemu()

#Session 2 [192.168.0.2]

InitNemu()
VRemote('line')
VSlirp('slirp')
VHost('b', hds=[VFs('freebsd.img', type='cow')]
      , nics=[VNic(hw='a0:00:00:00:00:10')
             , VNic(hw='a0:00:00:00:00:11')]
      , m=1024, cpu='SandyBridge')

SetIface('line', addr='192.168.0.1', port=7000)

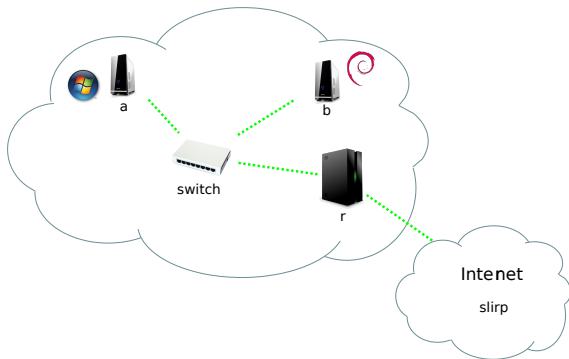
Link('b:0', 'line')
Link('b:1', 'slirp')

StartNemu()
```

VRouter

- **directement configuré** et géré par **NEmu** ;
- basé sur une version modifiée de **TinyCore Linux** ;
- fournit des **services réseaux** variés :
 - serveur DHCP ;
 - serveur DNS ;
 - serveur SSH ;
 - serveur HTTP ;
 - serveur NTP ;
 - utilitaires de QoS ;
 - pare-feu ;
 - protocoles de routage dynamique ;
 - etc.
- fournit un système de **plugin** simple dans le but de permettre l'ajout de **nouveaux services**.

NEmu en pratique - Scénario filaire domestique



```
InitNemu()
```

```
VSwitch('switch', niface=3)
```

```
VSlirp('slirp', net='192.168.0.0/24')
```

```
VHost('a', ...)
```

```
VHost('b', ...)
```

```
VRouter('r', nics=[VNic(), VNic()], services=[
    Service('ipforward'),
    Service('gateway', 0),
    Service('ifup', '1:192.168.0.1'),
    Service('dnsmasq', domain='home'
            , net='192.168.0.0/24'
            , start='192.168.0.10'
            , end='192.168.0.20'
            , ifaces=[1]),
    Service('masquerade'
            , ipsrc='192.168.0.0/24'
            , ifdst=0),
    Service('sshd')])
```

```
Link('a', 'switch:0')
```

```
Link('b', 'switch:1')
```

```
Link('r:1', 'switch:3')
```

```
Link('r:0', 'slirp')
```

```
StartNemu()
```

Architecture de NEmu - Lien virtuel mobile

Verrou technique

QEMU ne dispose **pas de carte** réseau virtuelle **sans fil**.

Solution

Créer un **équipement indépendant** connecté à QEMU par une **carte réseau filaire**.

Définition

Un **lien virtuel mobile (VAirLink)** est une **extension** des liens virtuels standards permettant d'émuler des **liens sans fil**.

Implémentation

vnd	{	VAirWic	<i>carte sans fil (ad-hoc ou infrastructure)</i>
		VAirAp	<i>point d'accès sans fil (infrastructure)</i>
		VAirRemote	<i>pont virtuel vers un équipement sans fil distant</i>

Architecture de NEmu - Lien virtuel mobile

Verrou technique

QEMU ne dispose **pas de carte** réseau virtuelle **sans fil**.

Solution

Créer un **équipement indépendant** connecté à QEMU par une **carte réseau filaire**.

Définition

Un **lien virtuel mobile (VAirLink)** est une **extension** des liens virtuels standards permettant d'émuler des **liens sans fil**.

Implémentation

vnd	{	VAirWic	<i>carte sans fil (ad-hoc ou infrastructure)</i>
		VAirAp	<i>point d'accès sans fil (infrastructure)</i>
		VAirRemote	<i>pont virtuel vers un équipement sans fil distant</i>

Architecture de NEmu - Lien virtuel mobile

Verrou technique

QEMU ne dispose **pas de carte** réseau virtuelle **sans fil**.

Solution

Créer un **équipement indépendant** connecté à QEMU par une **carte réseau filaire**.

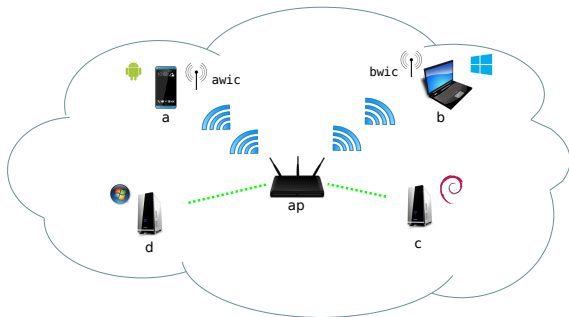
Définition

Un **lien virtuel mobile (VAirLink)** est une **extension** des liens virtuels standards permettant d'émuler des **liens sans fil**.

Implémentation

vnd	{	VAirWic	<i>carte sans fil (ad-hoc ou infrastructure)</i>
		VAirAp	<i>point d'accès sans fil (infrastructure)</i>
		VAirRemote	<i>pont virtuel vers un équipement sans fil distant</i>

NEmu en pratique - Scénario hybride domestique



```

InitNemu()

VAirAp('ap', niface=2)

VAirWic('awic')
VAirWic('bwic')
SetAirMode('awic', 'infra')
SetAirMode('bwic', 'infra')

VHost('a', ...)
VHost('b', ...)
VHost('c', ...)
VHost('d', ...)

Link('a', 'awic')
Link('b', 'bwic')
Link('c', 'ap:0')
Link('d', 'ap:1')

Join('awic', 'ap')
Join('bwic', 'ap')

ConfIface('awic:ap', bandwidth=54000000)
ConfIface('bwic:ap', direction='in'
          , bandwidth=54000000
          , ber=0.1)
ConfIface('bwic:ap', direction='out'
          , bandwidth=1048576)

StartNemu()

```

Architecture de NEmu - Mobilisateur

Définition

Un **mobilisateur** permet de faire **évoluer la connectivité** des **équipements sans fil** au cours du temps.

Implémentation

Scénario de mobilité : mobilité des éléments dans l'espace et dans le temps.



Scénario de connectivité : évolution de la connectivité des liens sans fil dans le temps.



Ordonnanceur temps réel : communique à NEmu les événements de connectivité.

Architecture de NEmu - Mobilisateur

Définition

Un **mobilisateur** permet de faire **évoluer la connectivité** des **équipements sans fil** au cours du temps.

Implémentation

Scénario de mobilité : mobilité des éléments dans l'espace et dans le temps.



Scénario de connectivité : évolution de la connectivité des liens sans fil dans le temps.



Ordonnanceur temps réel : communique à NEmu les événements de connectivité.

Architecture de NEmu - Mobilisateur

Définition

Un **mobilisateur** permet de faire **évoluer la connectivité** des **équipements sans fil** au cours du temps.

Implémentation

Scénario de **mobilité** : mobilité des éléments dans l'espace et dans le temps.



Scénario de **connectivité** : évolution de la connectivité des liens sans fil dans le temps.



Ordonnanceur temps réel : communique à NEmu les événements de connectivité.

Architecture de NEmu - Mobilisateur

Définition

Un **mobilisateur** permet de faire **évoluer la connectivité** des **équipements sans fil** au cours du temps.

Implémentation

Scénario de mobilité : mobilité des éléments dans l'espace et dans le temps.



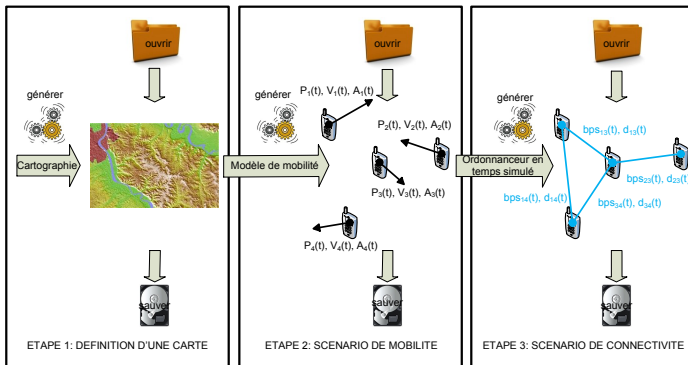
Scénario de connectivité : évolution de la connectivité des liens sans fil dans le temps.



Ordonnanceur temps réel : communique à NEmu les événements de connectivité.

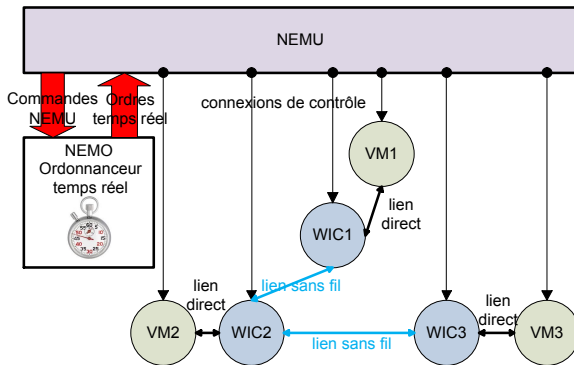
nemo (network mobilizer)

- initié pour **NEmu** en 2012 ;
- logiciel sous **LGPL**.

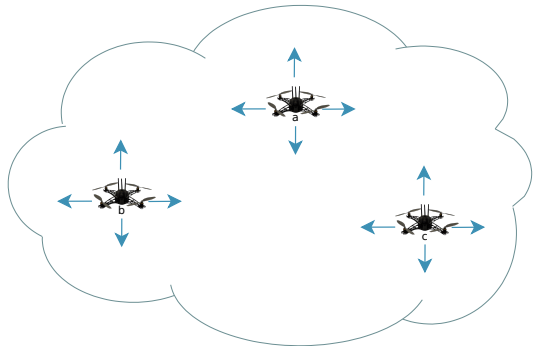


nemo (network mobilizer)

- initié pour **NEmu** en 2012 ;
- logiciel sous **LGPL**.



NEmu en pratique - Scénario mobile



```
InitNemu()  
  
VAirWic('awic')  
VAirWic('bwic')  
VAirWic('cwic')  
  
SetAirMode('awic', 'adhoc')  
SetAirMode('bwic', 'adhoc')  
SetAirMode('cwic', 'adhoc')  
  
VHost('a', ...)  
VHost('b', ...)  
VHost('c', ...)  
  
Link('a', 'awic')  
Link('b', 'bwic')  
Link('c', 'cwic')  
  
MobNemu('mob', nodes=['awic', 'bwic', 'cwic'])  
  
GenMobNemu('mob', width=1000, height=1000  
           , time=20, events=100)  
  
StartNemu()  
  
StartMobNemu('mob')
```


Futur

- **migration** de réseaux ;
- une interface **graphique** ;
- nouveaux **services** pour **VRouter** ;
- nouvelles **options** pour **vnd** :
 - spanning tree,
 - VLAN de types 2 et 3.
- nouvelles **options** pour **nemo** :
 - carte 3D,
 - obstacles,
 - modèles de mobilité plus complexes,
 - connectivité plus précise.

Questions



<http://nemu.valab.net>