

réseau

- tcp/ip est supporté depuis mathusalem par tous les Unix
- configuration:
 - adresse IP
 - routage
 - services réseau utilisés par la machine
 - services réseaux fournis par la machine

148

réseau : interface réseau

- une adresse ip peut-être affectée à chaque interface réseau
- nom des interfaces réseau
 - Linux: eth0, eth1, eth0:0 (alias: ràf)
 - OpenBSD, FreeBSD: nom spécifique au pilote de la carte (ex.: pcn0, vr0, fxp0, ...)
- interface spécifique:
 - interface de bouclage: lo sous Linux
 - liaison point à point, ppp, ...: ppp, tun0, ...
- le noyau doit contenir directement ou via modules:
 - le pilote de la carte
 - les pilotes des protocoles réseau utilisés

149

Configuration d'une interface réseau: ifconfig

- ifconfig: configurer une interface réseau
 - syntaxe dépendant de l'OS: ifconfig interface options
 - options:
 - up/down,
 - adresse ip, masque, mtu, ...
 - media (10/100/..., half/full duplex), adresse ethernet, ...
- ifconfig: exemples
 - ifconfig -a : affiche toutes les interfaces (+ informations)
 - ifconfig eth0 192.168.24.85 netmask 255.255.255.0
up: configure et active eth0

Configuration d'une interface réseau: via des scripts/fichiers de configuration

- Linux debian: /etc/network/interfaces: adresse IP, masque, ...

```
auto eth0
iface eth0 inet static
    address 195.221.165.248
    netmask 255.255.255.0
    network 195.221.165.0
    broadcast 195.221.165.255
    gateway 195.221.162.249
```

- OpenBSD: /etc/hostname.nomIF

```
inet 192.168.197.55 255.255.255.0 NONE
```

- FreeBSD: /etc/rc.conf

```
ifconfig_vx0="inet 195.159.221.165 netmask
255.255.255.0"
```

151

Configuration d'une interface réseau: Solaris

- /etc/hostname.nomIF: contient une entrée: le nom ou l'IP v4 de l'interface (notation CIDR acceptée)
- /etc/nodename: le nom du système local (non FQDN)
- /etc/defaultdomain: le nom de domaine complètement qualifié de la machine
- /etc/defaultrouter: contient les adresses des routeurs que l'hôte pourra utiliser.
- /etc/inet/hosts : hosts database (/etc/hosts est un lien symbolique vers ce fichier)

Références: System administration guide : IP services (voir sur <http://docs.sun.com>)

152

Etat d'une interface réseau

- ifconfig nomInterface

```
fxp0:
flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST>
mtu 1500
    inet 192.168.161.1 netmask 0xfffff00 broadcast
192.168.161.255
    inet6 fe80::2a0:c9ff:fe9e:dad2%fxp0 prefixlen 64
scopeid 0x1
    ether 00:a0:c9:9e:da:d2
    media: Ethernet autoselect (100baseTX <full-duplex>)
    status: active
```

- netstat -i:

```
$netstat -i -I fxp0
Name Mtu Network Address Ipkts Ierrs Opkts
Oerrs Coll
fxp0 1500 <Link#1> 00:a0:c9:9e:da:d2 918366 0 952442
0 0
fxp0 1500 192.168.161 192.168.161.1 1916 - 65737
- -
fxp0 1500 fe80:1::2a0 fe80:1::2a0:c9ff: 0 - 0 153
- -
```

test de connectivité: ping

- ping: envoie un paquet icmp echo request et attend un paquet icmp echo response
- si ça ne passe pas, il est possible que ça soit le paquet retour qui n'arrive pas
- test à compléter par une analyse de trames (tcpdump, ethereal, ...) pour voir où est le problème
- nmap, hping permet de faire de même via tcp ou udp en choisissant le port source (pour éviter certains filtres)
- arp: gestion du cache arp

154

Demo:

- demo où l'on teste la connectivité entre deux postes séparés par un routeur
- test entre les machines directement connectées
- test entre les deux machines extrêmes
- le second poste aura un routeur par défaut incorrect
 - les paquets ne revienne pas
 - mettre en évidence
 - que le paquet part (analyse de trame)
 - que la paquet arrive
 - que le paquet retour ne part pas (pb arp)

155

routage

- le routage permet à deux machines non directement reliées de communiquer via des machines intermédiaires appelés routeurs.
- un poste a en général une configuration simple: routeur par défaut
- cas plus complexes:
 - routage statique
 - routage dynamique (sort du contexte de cet enseignement)
- machine routeur:
 - accepte les paquets destinés à d'autres hôtes
 - le routage ip doit être activé

156

routage : configuration

- routes statiques: via la commande route ou fichier de configuration
- fichiers de configuration
 - Debian Gnu Linux:
 - /etc/network/interfaces : adresse IP, **routeur par défaut** & Co
 - debian: /etc/network/options: active le routage
 - ubuntu: /etc/sysctl.conf pour l'activation du routage
 - FreeBSD:
 - /etc/rc.conf: routeur par défaut, routes statiques
 - OpenBSD:
 - /etc/mygate: routeur par défaut

157

netstat -r: table de routage

- affiche la table de routage
 - une entrée pour chaque sous-réseau de chaque interface réseau (le champ passerelle est à 0.0.0.0)
 - une entrée pour le routeur par défaut (le champ destination est à 0.0.0.0)
 - une entrée par route statique.
- option -n : pas de conversion des valeurs numériques en valeurs littérales (évite l'utilisation du dns)

```
petit@sarge-test:~$ netstat -rn
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic  MSS Fenêtre  irtt Iface
192.168.100.0    192.168.244.60  255.255.255.0   UG     0 0         0 eth0
192.168.244.0    0.0.0.0         255.255.255.0   U      0 0         0 eth0
0.0.0.0          192.168.244.2  0.0.0.0         UG     0 0         0 eth0
```

Netstat

- obtenir des informations sur la configuration/les logiciels réseau d'un ordinateur
- des options dépendant du système d'exploitation
- exemple d'utilisation:
 - option commune: -n: désactive la résolution des adresses numériques (dns, ports, ...)
 - netstat -a: surveillance de l'état des connexion réseau
 - netstat -i : stat. trafic des interfaces réseau
 - netstat -r: table de routage
 - netstat -s: stat. par protocole tcp/ip

159

netstat -a: surveillance de l'état des connexion réseau

- « netstat -taupe » :
 - t: tcp
 - a, --all
 - u: udp
 - p: pid et programme auquel appartient la socket
 - e ou --extended (on peut aussi mettre -ee pour plus de détail)

160

netstat -s: stat. par protocole tcp/ip

161

services réseau, super-serveurs

- notion de socket
- numéros de ports
- démarrage via script
- démarrage via inetd/xinetd
- tcpd: tcp wrapper
- rpc et portmapper

162

inetd/xinetd

- pb: beaucoup de services potentiels qui ne servent pas tous ou rarement
- Solution pour les services réseau : on ne lance les services peu utilisés que lorsqu'une connexion se présente
- inetd: daemon qui gère les autres daemon
- inetd.conf:

```
ftp      stream  tcp      nowait  root    /usr/libexec/ftpd
ftpd -l

auth     stream  tcp      wait    root    /usr/local/sbin/identd identd -w -t120
pop3     stream  tcp      nowait  root    /usr/sbin/tcpd
         /usr/sbin/ipop3d
```

163

tcp wrappers

- But: interdire l'accès à des services en fonction de la machine demandeuse
- depuis inetd via tcpd
- via bibliothèque dynamique ad hoc
- rãf: la syntaxe du fichier de configuration sera détaillée dans la prochaine version de ce document

164

rpc/portmapper

- Principe:
 - un serveur qui démarre indique à portmap sur quel port il écoute et quel service il rend (/etc/rpc)
 - un client qui veut se connecter à un serveur demande au portmapper (port 111) sur quel port écoute le serveur qu'il veut joindre
- application: nfs, nis
- commandes utilisateur : rpcinfo
- rpc et sécurité
 - tcp-wrapper
 - fixer le port utilisé par les serveurs (nfs, nis le permettent)

165

Demo:

- donner un exemple de capture de trame avec nis ou nfs pour montrer le processus (rà: préciser le contexte de l'exemple dans la prochaine version de ce document)
- Le but est de montrer la connexion sur le port 111 (portmapper) pour trouver le port sur lequel écoute réellement le service.