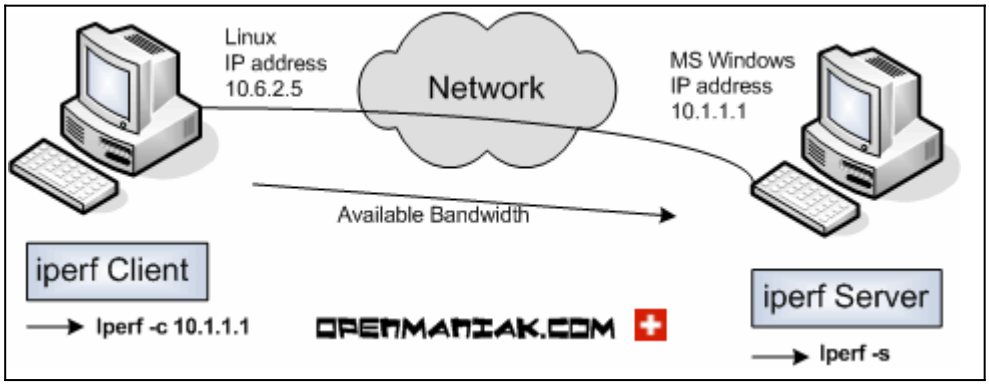
Iperf

Iperf est un outil pour mesurer la bande passante et la qualité d'un lien réseau. Ce dernier est délimité par deux machines sur lesquelles est installé Iperf.
La qualité d'un lien est déterminée principalement par les facteurs suivants:
- Latence (temps de réponse ou RTT): peut être mesurée à l'aide d'un Ping.
- Gigue ou jitter en anglais (variation de la latence): peut être mesurée par un test Iperf UDP.
- Perte de paquet: peut être mesurée avec un test Iperf UDP.

Quant à la bande passante, elle est mesurée par des tests TCP.
Pour être simple la différence entre TCP (Transmission Control Protocol) et UDP (User Datagram Protocol) est que TCP utilise des processes pour vérifier que les paquets sont correctement envoyés au receveur. ceci n'est pas le cas pour UDP où les paquets sont envoyés sans aucune vérification mais avec l'avantage d'être plus rapide que TCP.
Iperf utilise les différentes propriétés de TCP et d'UDP pour fournir des statistiques sur des liens réseaux.
Iperf peut être installé très facilement sur n'importe quel système UNIX/Linux ou Microsoft Windows.
Un hôte doit être configuré en tant que client et l'autre en tant que serveur.

Voici un petit labo où Iperf est installé sur une machine Linux et Microsoft Windows.
Linux est utilisé comme client Iperf et Windows comme serveur Iperf. Bien sûr, il est aussi possible d'utiliser deux machines Linux.



Tests Iperf:

défaut	Paramètres par défaut
-f	Formatage des données
-r	Bande passante bidirectionnelle
-d	Bande passante bidirectionnelle simultanée
-w	Taille de la fenêtre TCP
-h	Aide
-p, -t, -i	Port, temps et intervalle
-u, -b	Tests UDP, configuration bande passante
-m	Affichage de la taille de segment maximale
-M	Configuration de la taille de segment maximale
-P	Tests parallèles

Jperf:

Pas d'argument	Paramètres par défaut
-d	Bande passante bidirectionnelle simultanée
-u, -b	Tests UDP, configuration bande passante

Paramétrage Iperf par défaut:
Voir la section "Jperf"

Par défaut, le client Iperf se connecte au serveur Iperf sur le port TCP 5001 et la bande passante affichée par Iperf est celle du client au serveur.
Si vous voulez utiliser des tests UDP, utilisez l'argument -u.
Les arguments client Iperf-d et -r mesurent les bandes passantes bidirectionnelles. (Voir plus loin dans ce tutorial)

- Côté client:

#iperf -c 10.1.1.1

Client connecting to 10.1.1.1, TCP port 5001
TCP window size: 16384 Byte (default)
---
[3] local 10.6.2.5 port 33453 connected with 10.1.1.1 port 5001
[3] 0.0-10.2 sec 1.26 MBytes 1.05 Mbits/sec

- Côté serveur:

#iperf -s

Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)
---
[852] local 10.1.1.1 port 5001 connected with 10.6.2.5 port 33453
[ID] Interval Transfer Bandwidth
[852] 0.0-10.6 sec 1.26 MBytes 1.03 Mbits/sec

Formatage des données: (argument -f)

L'argument -f permet d'afficher les résultats selon le format désiré: bits(b), bytes(B), kilobits(k), kilobytes(K), megabits(m), megabytes(M), gigabits(g) ou gigabytes(G).
Généralement, les mesures de bande passante sont affichées en bits/sec (ou Kilobits/sec, etc ...) et une quantité de données est affichée en octets (or Kilobytes, etc ...).
Pour mémoire, 1 octet (byte en anglais) est égal à 8 bits et dans le domaine informatique 1 kilo est égale à 1024 (2^10).
Par exemple: 100'000'000 octets ne sont pas égals à 100 Mbytes mais à 100'000'000/1024/1024 = 95.37 Mbytes.

- Côté client:

#iperf -c 10.1.1.1 -f b

Client connecting to 10.1.1.1, TCP port 5001
TCP window size: 16384 Byte (default)

---

[3] local 10.6.2.5 port 54953 connected with 10.1.1.1 port 5001
[3] 0.0-10.2 sec 1359872 Bytes 1064272 bits/sec

- Côté serveur:

#iperf -s

Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)

---

[852] local 10.1.1.1 port 5001 connected with 10.6.2.5 port 33453
[ID] Interval Transfer Bandwidth
[852] 0.0-10.6 sec 920 KBytes 711 Kbits/sec

- Mesure de la bande passante bidirectionnelle: (argument -r)

Le serveur Iperf se connecte en retour sur le client permettant la mesure de la bande passante bidirectionnelle. Par défaut, seule la bande passante du client au serveur est mesurée.
Si vous voulez mesurer la bande passante bidirectionnelle de manière simultanée, utilisez l'argument -d. (Voir le test suivant)

- Côté client:

#iperf -c 10.1.1.1 -r

Server listening on TCP port 5001

TCP window size: 85.3 KByte (default)

Client connecting to 10.1.1.1, TCP port 5001
TCP window size: 16.0 KByte (default)
---
[5] local 10.6.2.5 port 35726 connected with 10.1.1.1 port 5001
[5] 0.0-10.0 sec 1.12 MBytes 936 Kbits/sec
[4] local 10.6.2.5 port 5001 connected with 10.1.1.1 port 1640
[4] 0.0-10.1 sec 74.2 MBytes 61.7 Mbits/sec

- Côté serveur:

#iperf -s

Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)
---
[852] local 10.1.1.1 port 5001 connected with 10.6.2.5 port 54355
[ID] Interval Transfer Bandwidth
[852] 0.0-10.1 sec 1.15 MBytes 956 Kbits/sec

Client connecting to 10.6.2.5, TCP port 5001
TCP window size: 8.00 KByte (default)
---
[824] local 10.1.1.1 port 1646 connected with 10.6.2.5 port 5001
[ID] Interval Transfer Bandwidth
[824] 0.0-10.0 sec 73.3 MBytes 61.4 Mbits/sec

Haut de la page

- Mesure de la bande passante bidirectionnelle simultanée: (argument -d)

Voir la section "Jperf".

Pour mesurer les bandes passantes bidirectionnelles simultanées, utilisez l'argument -d. Si vous voulez tester les bandes passantes séquentiellement, utilisez l'argument -r (Voir le test précédent).
Par défaut (c'est-à-dire sans les arguments -r ou -d), seule la bande passante du client au serveur est mesurée.

- Côté client:

#iperf -c 10.1.1.1 -d

--
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
--
--
Client connecting to 10.1.1.1, TCP port 5001
TCP window size: 16.0 KByte (default)
--
[5] local 10.6.2.5 port 60270 connected with 10.1.1.1 port 5001
[4] local 10.6.2.5 port 5001 connected with 10.1.1.1 port 2643
[4] 0.0-10.0 sec 76.3 MBytes 63.9 Mbits/sec
[5] 0.0-10.1 sec 1.55 MBytes 1.29 Mbits/sec

➙ Côté serveur:

#iperf -s
--
Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)
--
[852] local 10.1.1.1 port 5001 connected with 10.6.2.5 port 60270
--
Client connecting to 10.6.2.5, TCP port 5001
TCP window size: 8.00 KByte (default)
--
[800] local 10.1.1.1 port 2643 connected with 10.6.2.5 port 5001

[ID]	*Interval*	*Transfer*	*Bandwidth*
[800]	*0.0-10.0 sec*	*76.3 MBytes*	*63.9 Mbits/sec*
[852]	*0.0-10.1 sec*	*1.55 MBytes*	*1.29 Mbits/sec*

Haut de la page

Taille de la fenêtre TCP: (argument -w)

La taille de la fenêtre TCP corresponds aux données qui peuvent être mise en tampon pendant une connexion sans la validation du receveur.
Elle est comprise entre 2 et 65535 bytes.
Sur les systèmes Linux, quand on spécifie une taille de fenêtre TCP avec l'argument -w, le noyau alloue le double de la valeur indiquée.

➙ Côté client:

#iperf -c 10.1.1.1 -w 2000
WARNING: TCP window size set to 2000 bytes. A small window size
will give poor performance (une petite taille de fenêtre donnera de faible performances). *See the Iperf documentation.*

Client connecting to 10.1.1.1, TCP port 5001
TCP window size: 3.91 KByte (WARNING: requested 1.95 KByte)
---
[3] local 10.6.2.5 port 51400 connected with 10.1.1.1 port 5001
[3] 0.0-10.1 sec 704 KBytes 572 Kbits/sec

→ Côté serveur:

#iperf -s -w 4000

Server listening on TCP port 5001
TCP window size: 3.91 KByte
---
[852] local 10.1.1.1 port 5001 connected with 10.6.2.5 port 51400
[ID] Interval Transfer Bandwidth
[852] 0.0-10.1 sec 704 KBytes 570 Kbits/sec

Haut de la page

■ Port de communication (-p), temps (-t) et intervalle (-i):

Le port de communication du serveur Iperf peut être changé avec l'argument -p. Il doit être configuré sur le client et le serveur avec la même valeur, par défaut le port TCP 5001.
L'argument -t spécifie la durée du test en seconde, par défaut 10 secondes.
L'argument -i indique l'intervalle en seconde entre les rapports périodiques de bande passante.

→ Côté client:

#iperf -c 10.1.1.1 -p 12000 -t 20 -i 2

Client connecting to 10.1.1.1, TCP port 12000
TCP window size: 16.0 KByte (default)
---
[3] local 10.6.2.5 port 58316 connected with 10.1.1.1 port 12000
[3] 0.0- 2.0 sec 224 KBytes 918 Kbits/sec
[3] 2.0- 4.0 sec 368 KBytes 1.51 Mbits/sec
[3] 4.0- 6.0 sec 704 KBytes 2.88 Mbits/sec

[3] 6.0- 8.0 sec 280 KBytes 1.15 Mbits/sec
[3] 8.0-10.0 sec 208 KBytes 852 Kbits/sec
[3] 10.0-12.0 sec 344 KBytes 1.41 Mbits/sec
[3] 12.0-14.0 sec 208 KBytes 852 Kbits/sec
[3] 14.0-16.0 sec 232 KBytes 950 Kbits/sec
[3] 16.0-18.0 sec 232 KBytes 950 Kbits/sec
[3] 18.0-20.0 sec 264 KBytes 1.08 Mbits/sec
[3] 0.0-20.1 sec 3.00 MBytes 1.25 Mbits/sec

→ Côté serveur:

#iperf -s -p 12000

Server listening on TCP port 12000
TCP window size: 8.00 KByte (default)

---

[852] local 10.1.1.1 port 12000 connected with 10.6.2.5 port 58316
[ID] Interval Transfer Bandwidth
[852] 0.0-20.1 sec 3.00 MBytes 1.25 Mbits/sec

Haut de la page

■ Tests UDP (-u), paramétrage de la bande passante (-b)
Voir la section "Jperf".

Les tests UDP avec l'argument -u vont donner de précieuses informations sur la gigue (jitter en anglais) ou les pertes de paquets. Si vous ne spécifiez pas l'argument -u, Iperf utilise TCP.
Pour garder une bonne qualité de lien, la perte de paquet ne devrait pas dépasser l%. Un haut taux de perte de paquet générera un grand nombre de retransmissions de segments TCP ce qui affectera la bande passante.

La Gigue (jitter) est basiquement la variation de la latence et ne dépends pas de cette dernière. Il est tout à fait possible d'avoir des temps de réponse élevés et une gigue très basse. La valeur de la gigue est particulièrement importante pour des liens réseaux supportant la voix sur IP (VoIP, Voice over IP) parce qu'une gigue élevée peut interrompre un appel téléphonique.
L'argument -b permet d'allouer la bande passante désirée.

→ Côté client:

#iperf -c 10.1.1.1 -u -b 10m

Client connecting to 10.1.1.1, UDP port 5001

Sending 1470 byte datagrams
UDP buffer size: 108 KByte (default)
---
[3] local 10.6.2.5 port 32781 connected with 10.1.1.1 port 5001
[3] 0.0-10.0 sec 11.8 MBytes 9.89 Mbits/sec
[3] Sent 8409 datagrams
[3] Server Report:
[3] 0.0-10.0 sec 11.8 MBytes 9.86 Mbits/sec 2.617 ms 9/ 8409 (0.11%)

Côté serveur:

#iperf -s -u -i 1

Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 8.00 KByte (default)
---
[904] local 10.1.1.1 port 5001 connected with 10.6.2.5 port 32781

[ID]	*Interval*	*Transfer*	*Bandwidth*	*Jitter*	*Lost/Total Datagrams*
[904]	*0.0- 1.0 sec*	*1.17 MBytes*	*9.84 Mbits/sec*	*1.830 ms*	*0/ 837 (0%)*
[904]	*1.0- 2.0 sec*	*1.18 MBytes*	*9.94 Mbits/sec*	*1.846 ms*	*5/ 850 (0.59%)*
[904]	*2.0- 3.0 sec*	*1.19 MBytes*	*9.98 Mbits/sec*	*1.802 ms*	*2/ 851 (0.24%)*
[904]	*3.0- 4.0 sec*	*1.19 MBytes*	*10.0 Mbits/sec*	*1.830 ms*	*0/ 850 (0%)*
[904]	*4.0- 5.0 sec*	*1.19 MBytes*	*9.98 Mbits/sec*	*1.846 ms*	*1/ 850 (0.12%)*
[904]	*5.0- 6.0 sec*	*1.19 MBytes*	*10.0 Mbits/sec*	*1.806 ms*	*0/ 851 (0%)*
[904]	*6.0- 7.0 sec*	*1.06 MBytes*	*8.87 Mbits/sec*	*1.803 ms*	*1/ 755 (0.13%)*
[904]	*7.0- 8.0 sec*	*1.19 MBytes*	*10.0 Mbits/sec*	*1.831 ms*	*0/ 850 (0%)*
[904]	*8.0- 9.0 sec*	*1.19 MBytes*	*10.0 Mbits/sec*	*1.841 ms*	*0/ 850 (0%)*
[904]	*9.0-10.0 sec*	*1.19 MBytes*	*10.0 Mbits/sec*	*1.801 ms*	*0/ 851 (0%)*
[904]	*0.0-10.0 sec*	*11.8 MBytes*	*9.86 Mbits/sec*	*2.618 ms*	*9/ 8409 (0.11%)*

Haut de la page

Affichage de la taille de segment maximale (argument -m):

La taille de segment maximale (ou en anglais, Maximum Segment Size, MSS) est la plus grande quantité de données, en octets (bytes), qu'un ordinateur peut supporter en un unique et non-fragmenté segment TCP.
Elle peut être calculée de la manière suivante:
MSS = MTU - en-têtes (headers) TCP & IP
Les en-têtes TCP & IP occupent 40 octets.

La MTU (Maximum Transmission Unit, unité de transmission maximale) est la plus grande quantité de données qui peut être transférée dans une trame.
Voici quelques tailles de MTU par défaut pour des topologies réseaux différentes:
Ethernet - 1500 octets: utilisé dans un réseau local (LAN).
PPPoE - 1492 octets: utilisé sur des liens ADSL.
Token Ring (16Mb/sec) - 17914 octets: vielle technologie crée par IBM.
Connexion téléphonique - 576 octets

Généralement, une MTU (et une MSS) élevée permet une plus grande bande passante.

- Côté client:

#iperf -c 10.1.1.1 -m
--
Client connecting to 10.1.1.1, TCP port 5001
TCP window size: 16.0 KByte (default)
--
[3] local 10.6.2.5 port 41532 connected with 10.1.1.1 port 5001
[3] 0.0-10.2 sec 1.27 MBytes 1.04 Mbits/sec
[3] MSS size 1448 bytes (MTU 1500 bytes, ethernet)

Ici la MSS n'est pas égale à 1500 -40 mais à 1500 - 40 - 12(option Timestamps) = 1448

- Côté serveur:

#iperf -s
Haut de la page

- Configuration de la taille de segment maximale (argument -M):

Utilisez l'argument -M pour changer la taille de segment maximale. (Maximale Segment Size, MSS)
(Voir le test précédent pour plus d'explication sur la MSS)

#iperf -c 10.1.1.1 -M 1300 -m
WARNING: attempt to set TCP maximum segment size to 1300, but got 536
--
Client connecting to 10.1.1.1, TCP port 5001
TCP window size: 16.0 KByte (default)
--
[3] local 10.6.2.5 port 41533 connected with 10.1.1.1 port 5001
[3] 0.0-10.1 sec 4.29 MBytes 3.58 Mbits/sec
[3] MSS size 1288 bytes (MTU 1328 bytes, unknown interface)

→ Côté serveur:

#iperf -s

Haut de la page

Tests en parallèle (argument -P):

Utilisez l'argument -P pour lancer des tests en parallèle.

→ Côté client:

#iperf -c 10.1.1.1 -P 2

```
------------------------------------------------------------
Client connecting to 10.1.1.1, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[ 3] local 10.6.2.5 port 41534 connected with 10.1.1.1 port 5001
[ 4] local 10.6.2.5 port 41535 connected with 10.1.1.1 port 5001
[ 4]  0.0-10.1 sec  1.35 MBytes  1.12 Mbits/sec
[ 3]  0.0-10.1 sec  1.35 MBytes  1.12 Mbits/sec
[SUM] 0.0-10.1 sec  2.70 MBytes  2.24 Mbits/sec
```

→ Côté serveur:

#iperf -s

Haut de la page

Aide Iperf:

#iperf -h

```
Usage: iperf [-s|-c host] [options]
iperf [-h|--help] [-v|--version]

Client/Server:
-f  --format      [kmKM]  format to report: Kbits, Mbits, KBytes, MBytes
                          format de rapport: Kbits, Mbits, KBytes, MBytes
-i  --interval    #       seconds between periodic bandwidth reports
                          secondes entre les rapports périodiques de bande passante
-l  --len         #[KM]   length of buffer to read or write (default 8 KB)
                          longueur du tampon pour lire ou écrire (défaut 8 Kb)
-m  --print_mss           print TCP maximum segment size (MTU - TCP/IP header)
```

affiche la taille de segment TCP maximale (MTU - en-têtes TCP/IP)

-p	*--port*	*#*	*server port to listen on/connect to* port du serveur
-u	*--udp*		*use UDP rather than TCP* utilisation d'UDP plutôt que TCP
-w	*--window*	*#[KM]*	*TCP window size (socket buffer size)* taille de la fenêtre TCP (taille du tampon de la socket)
-B	*--bind*	*"host"*	*bind to "host", an interface or multicast address* attachement à l'"host" (hôte), une interface ou adresse multicast
-C	*--compatibility*		*for use with older versions does not sent extra msgs* pour l'utilisation avec de vieilles version
-M	*--mss*	*#*	*set TCP maximum segment size (MTU - 40 bytes)* configure le taille de segment TCP maximale (MTU - 40 bytes)
-N	*--nodelay*		*set TCP no delay, disabling Nagle's Algorithm* configure le paramètre "TCP no delay", désactivation de l'algorithme de Nagle.
-V	*--IPv6Version*		*Set the domain to IPv6* Configure le domaine en IPv6

Server specific:

-s	*--server*	*run in server mode* lancement en mode serveur
-U	*--single_udp*	*run in single threaded UDP mode* -
-D	*--daemon*	*run the server as a daemon* lancement du serveur en démon m

Client specific:

-b	*--bandwidth*	*#[KM]*	*for UDP, bandwidth to send at in bits/sec (default 1 Mbit/sec, implies -u)* pour UDP, bande passante à envoyer en bits/sec (défaut 1 Mbit/sec, implique -u)
-c		*"host"*	*run in client mode, connecting to "host"*
	--client		lancement en mode client, connexion à l'"host" (hôte).
-d			*Do a bidirectional test simultaneously*
	--dualtest		Fait un test bidirectionnel simultané.
-n		*#[KM]*	*number of bytes to transmit (instead of -t)*
	--num		nombre de bytes à transmettre (au lieu de -t)
-r			*Do a bidirectional test individually*
	--tradeoff		Fait un test bidirectionnel individuellement
-t		*#*	*time in seconds to transmit for (default 10 secs)*
	--time		temps en seconde pour transmettre (défaut 10 sec)
-F		*"name"*	*input the data to be transmitted from a file*
	--fileinput		-
-I			*input the data to be transmitted from stdin*
	--stdin		-
-L		*#*	*port to recieve bidirectional tests back on*
	--listenport		port pour recevoir des tests bidirectionnels en retour.
-P		*#*	*number of parallel client threads to run*
	--parallel		nombre de tests client en parallèle à lancer
-T		*#*	*time-to-live, for multicast (default 1)*

--ttl	time-to-live (temps de vie), pour multicast (défaut 1)

Miscellaneous:

-h --help	*print this message and quit* affiche ce message et quitte
-v --version	*print version information and quit* affiche l'information de version et quitte

Haut de la page

JPERF

Jperf est une interface graphique pour Iperf écrit en Java.

---

1. Installation:

Download Jperf.

Linux

Décompressez le fichier téléchargé:

#tar -xvf jperf2.0.0.zip
Lancez Jperf.

#cd jperf2.0.0
#./jperf.sh
Si vous avez le message suivant, ceci signifie que vous avez besoin d'installer Iperf avec la commande: "apt-get install iperf"

Iperf is probably not in your Path!
Please download it here 'http://dast.nlanr.net/Projects/Iperf/'
and put the executable in your PATH environment variable.

Microsoft Windows

Décompressez le fichier téléchargé avec votre programme favori.

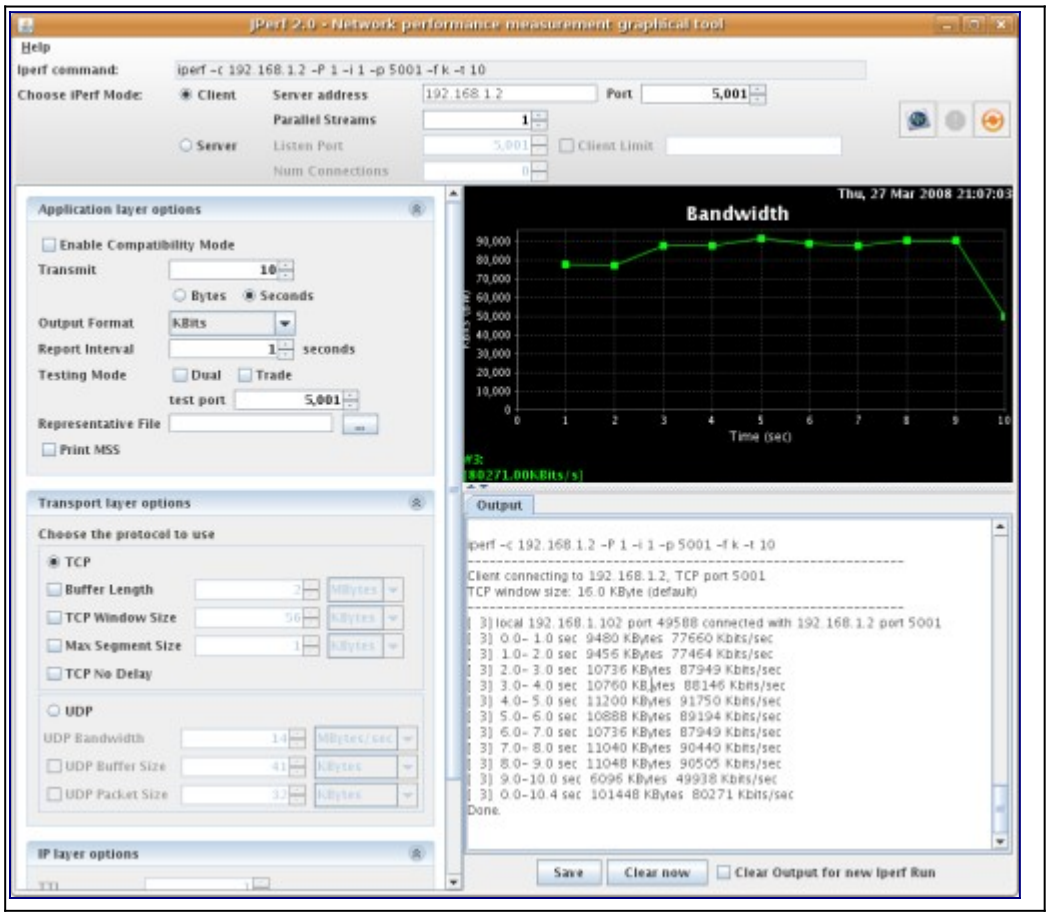
Accédez au dossier décompressé appelé par défaut "jperf2.0.0" et double-cliquez sur "iperf.bat".
Notez que l'utilitaire Iperf est déjà présent dans le dossier /bin.

---

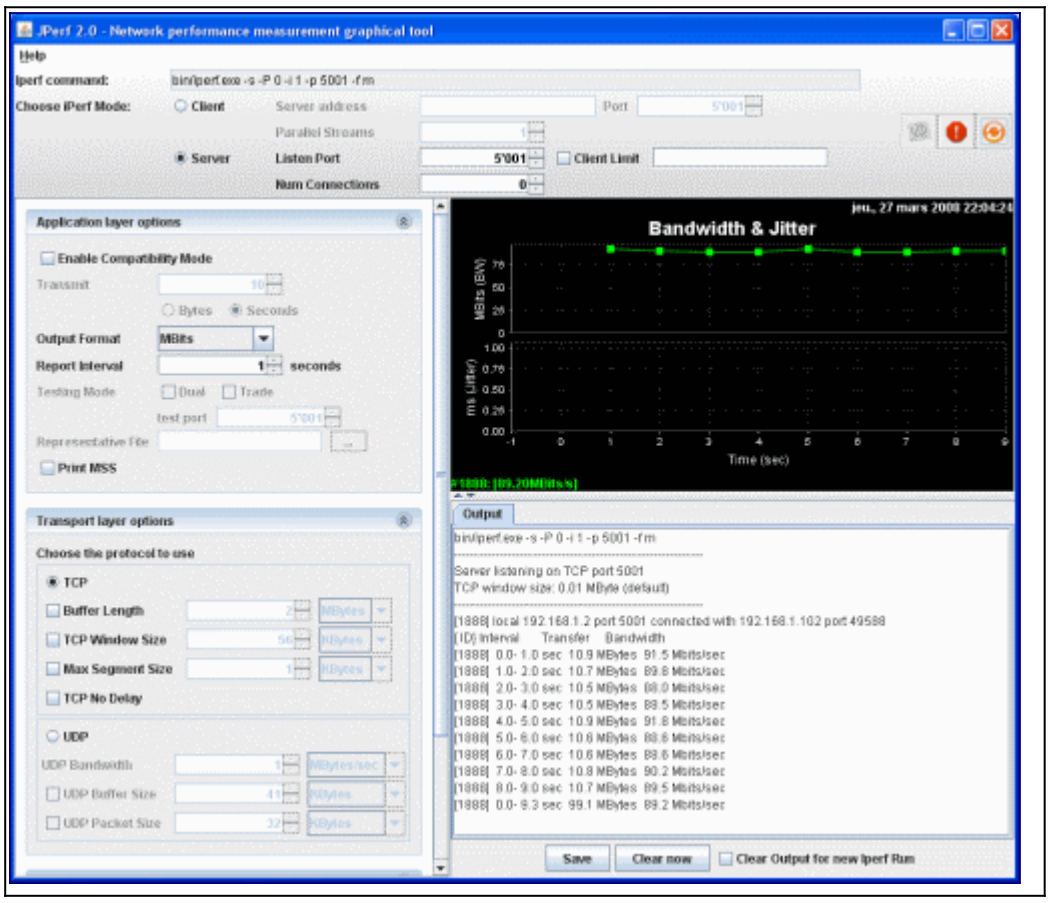
2. Exemples:

→ Paramètres par défaut, mesure de la bande passante:
Voir la section "Iperf" pour plus de détails.

- Client Linux:



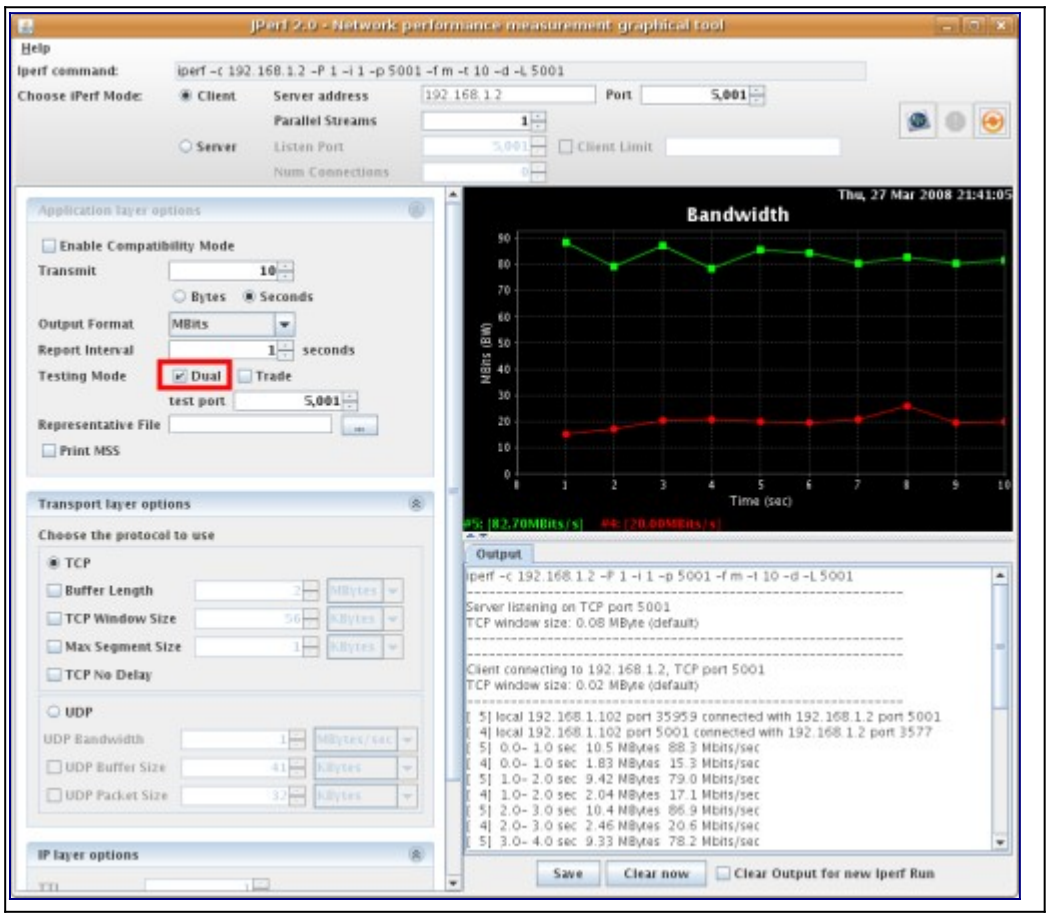
- Serveur Windows:



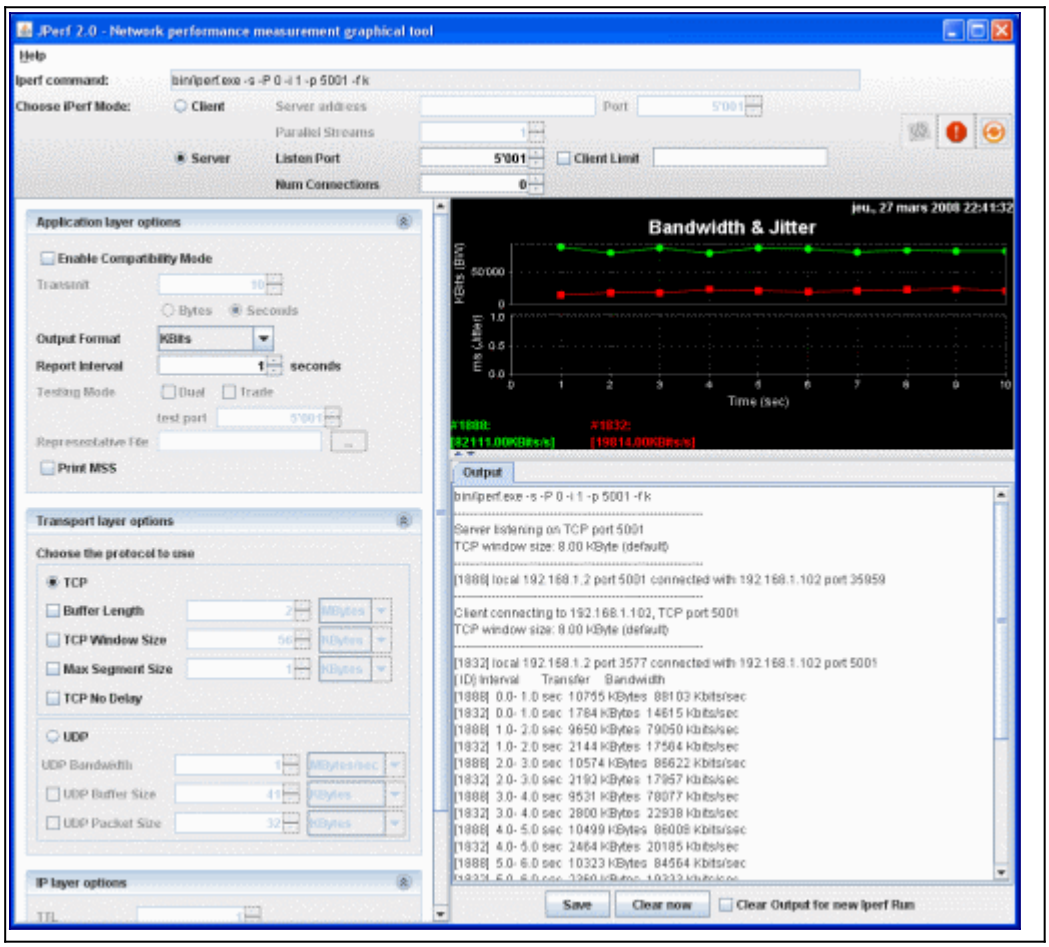
Haut de la page Jperf

→ Mesure de la bande passante bidirectionnelle simultanée:
Voir la section "Iperf" pour plus de détails.

- Client Linux:



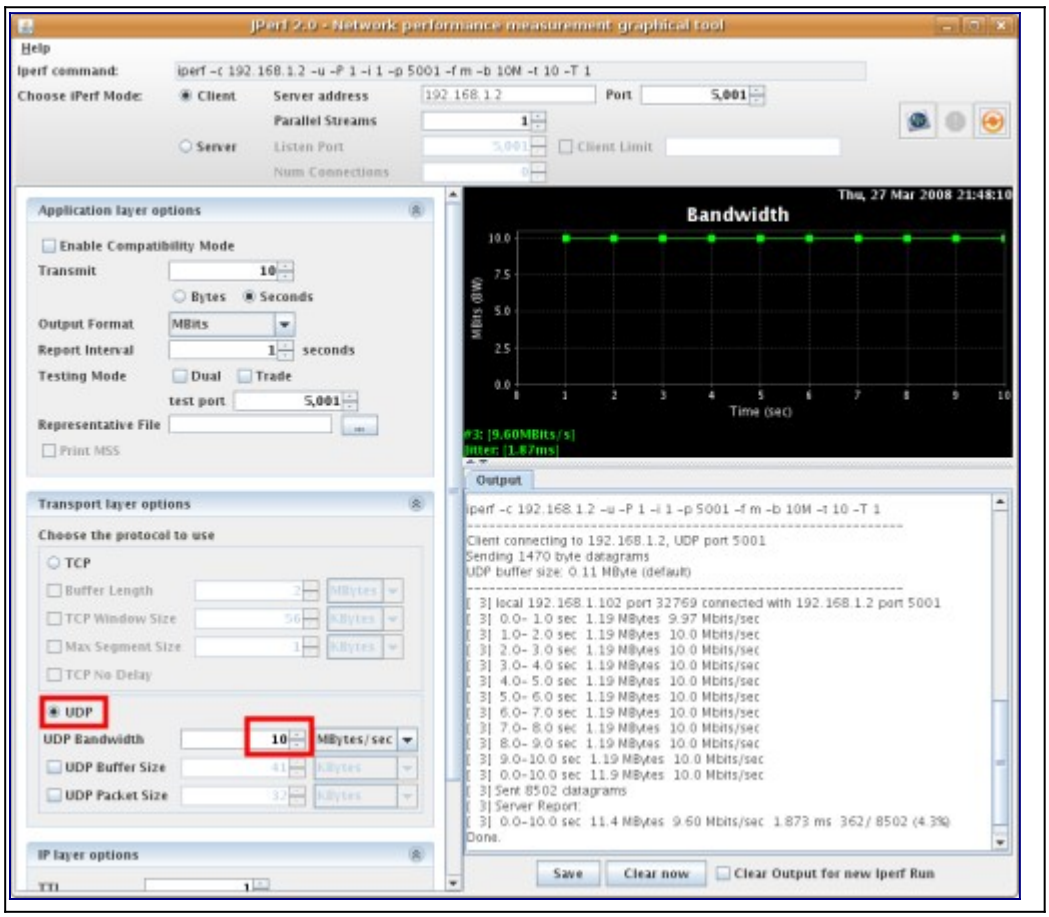
- Serveur Windows:



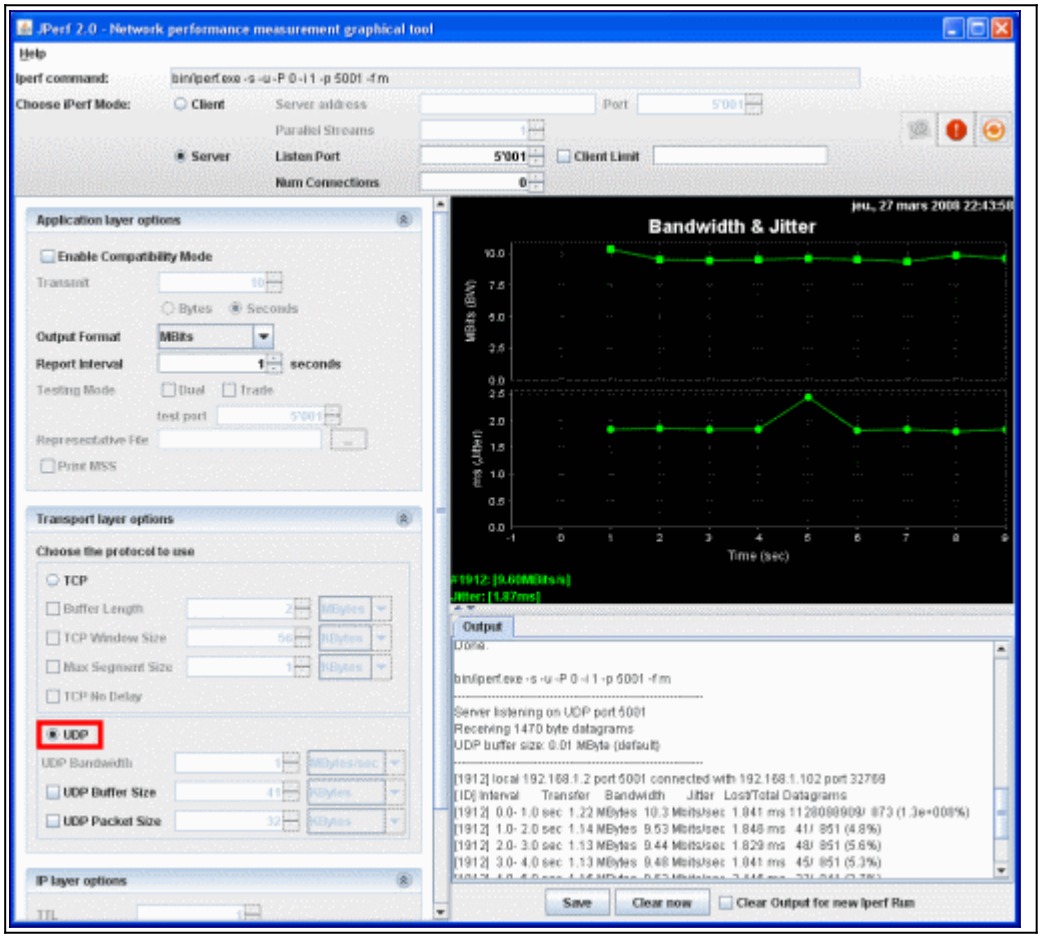
Haut de la page Jperf

→ Mesure de la gigue:
Voir la section "Iperf" pour plus de détails.

- Client Linux:



- Serveur Windows:


JPerf 2.0 - Network performance measurement graphical tool
Help
Iperf command:
bin\iperf.exe -s -u -P 0 -i 1 -p 5001 -f m
Choose iPerf Mode:
Client
Server address
Port
Parallel Streams
Server
Listen Port
5001
Client Limit
Num Connections
0
Application layer options
Enable Compatibility Mode
Transmit
Bytes
Seconds
Output Format
MBits
Report Interval
1
seconds
Testing Mode
Dual
Trade
test port
Representative File
Print MSS
Transport layer options
Choose the protocol to use
TCP
Buffer Length
TCP Window Size
Max Segment Size
TCP No Delay
UDP
UDP Bandwidth
UDP Buffer Size
UDP Packet Size
IP layer options
TTL
jeu., 27 mars 2008 22:43:58
Bandwidth & Jitter
MBits (BW)
ms (Jitter)
Time (sec)
Output
Done.
bin\iperf.exe -s -u -P 0 -i 1 -p 5001 -f m
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
[1912] local 192.168.1.2 port 5001 connected with 192.168.1.102 port 32769
[ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
[1912] 0.0- 1.0 sec 1.22 MBytes 10.3 MBits/sec 1.041 ms 5128088909/ 873 (1.3e+008%)
[1912] 1.0- 2.0 sec 1.14 MBytes 9.53 MBits/sec 1.848 ms 41/ 851 (4.8%)
[1912] 2.0- 3.0 sec 1.13 MBytes 9.44 MBits/sec 1.829 ms 48/ 851 (5.6%)
[1912] 3.0- 4.0 sec 1.13 MBytes 9.48 MBits/sec 1.041 ms 45/ 851 (5.3%)
Save
Clear now
Clear Output for new Iperf Run