

## Measuring Network Performance: Collisions Rate and Network Load

Assist. Mădălina MLAK

Informatics in Economy Department, Academy of Economic Studies Bucharest

*The article present which parameters affect the basic network performance and which tools can best be used for these measurements and measure these parameters. Various operating system commands can be used for measuring.*

**Keywords:** network, performance, bandwidth, throughput, measuring for network.

### Introduction

Network supports your applications and if your network's performance decreases, the execution of applications will be close behind. Network performance is a complex issue, but for the user a definition of network performance is measuring the speed of the network. Network bandwidth is a factor that determines the "speed" of a network as perceived by the end user. The measure of computer network performance is commonly given in units of bits per second (bps).

### 2. Factors which affecting network performance

Performance management helps network administrators achieve new world business objectives and consistent network availability and productivity. Performance problems are often observed immediately following some change to system hardware and software. Many of factors which affecting overall network performance can be caused by the:

- network bandwidth;
- server bandwidth;
- server CPU workload;
- server memory;
- server network interface;
- client bandwidth;
- client CPU workload;
- client memory;
- client network interface.

Those factors can be analyzed using operating system's commands.

### 3. Commands for measuring network performance

The first step in measuring network performance is to determine if the network even working.

With ping you can verify the availability of the network.

Receiving an echo ping reply packet from the remote system means that the network is available between the client and server devices.

By selecting different remote systems on the network, you can test if all the segments on the network are available. After that you can measure network performance.

Most of the existing commercial tools for improving network performance require large amounts of knowledge. But the operating system commands are easy to use and interpret.

Several commands can measure the network and offer many information and statistics, but only a part of this information is related to network performance. Use the following commands to collect data for each of the factors.

For:

- measuring network bandwidth we can use **netperf**.
- server bandwidth and client bandwidth we can use **ttcp** between two stations in a LAN (one is server and other is client and vice versa).
- server CPU workload and client CPU workload we can use commands: **sar**, **vmstat** and **iostat**. The command is executed for each system.
- server memory and client memory we can use commands: **vmstat** and **netstat** with option **-m**;
- server network interface and client network interface we can use commands: **ifconfig** and **netstat**.

Note: Is possible to have different output format of those commands, depending on options and type of operating system.

### 3.1. Measuring network bandwidth

Bandwidth is a measure of the theoretical total capability of a network. Throughput measures the results. Bandwidth measures a network only, where throughput measures not only the network, the computers both hardware and software.

Measuring network bandwidth is necessary to determine the maximum throughput for

the system is capable with the installed network hardware and software.

Netperf is a benchmark utility that can be used to measure various aspects of networking performance of many different types of networking. It provides tests for both one direction throughput, and end-to-end latency.

In the following example for netperf command [4], we can observe the UDP traffic sent over a Gigabit Ethernet connection and for TCP stream test and the value for throughput (highlight):

```
atc03(/opt/netperf) -> netperf -fM -Hatc01-ge -tUDP_STREAM -- -m1472
UDP UNIDIRECTIONAL SEND TEST to atc01-ge
Socket Message Elapsed      Messages
Size  Size  Time      Okay Errors
bytes bytes secs      #      #
-----
65535 1472 10.01    527813 0
65535 1472 10.01    527719 0
74.05
74.04

atc03(/opt/netperf) -> netperf -fM -Hatc01 -tTCP_STREAM -- -s65536 -S65536
TCP STREAM TEST to atc01
Recv Send  Send
Socket Socket Message Elapsed
Size  Size  Size  Time      Throughput
bytes bytes bytes secs.    MBytes/sec
-----
65536 65536 65536 10.02    11.31

atc03(/opt/netperf) -> netperf -fM -Hros87252.rose.hp.com -tTCP_STREAM
TCP STREAM TEST to ros87252.rose.hp.com
Recv Send  Send
Socket Socket Message Elapsed
Size  Size  Size  Time      Throughput
bytes bytes bytes secs.    MBytes/sec
-----
32768 32768 32768 10.03    1.98
```

### 3.2. Monitoring server bandwidth and client bandwidth

Ttcp sending TCP traffic from client to server and from server to client. Options `-r` receive data, option `-t` transmit data.

For the first station the command is: **ttcp - r**

```
ttcp-r:          buflen=8192,          nbuf=2048,
align=16384/0, port=5001 tcp
ttcp-r: socket  ttcp-r: accept  from
193.226.34.186
ttcp-r: 16777216 bytes in 19.99 real seconds
= 819.60 KB/sec +++
ttcp-r: 10288 I/O calls, msec/call = 1.99,
calls/sec = 514.67
```

```
ttcp-r: 0.1user 3.4sys 0:19real 17%
```

For the second station the command is: **ttcp - t first station**

```
ttcp-t:          buflen=8192,          nbuf=2048,
align=16384/0, port=5001 tcp -> venus
ttcp-t: socket
ttcp-t: connect
ttcp-t: 16777216 bytes in 19.98 real seconds
= 820.02 KB/sec +++
ttcp-t: 2048 I/O calls, msec/call = 9.99,
calls/sec = 102.50
```

```
ttcp-t: 0.0user 2.3sys 0:19real 12%
```

The statistics for bandwidth are written bold. The both stations (client and server) are direct connected.

### 3.3. Monitoring server CPU workload and client CPU workload

For performance management, you must ensure that the device and network have sufficient CPU to function at all times.

Command `sar -q` displays the activity of CPU.

```
# sar -q 5 3
19:31:42 runq-sz %runocc swpq-sz %swpocc
19:31:47 1.0 100 1.0 100
19:31:52 2.0 100 1.0 100
19:31:57 1.0 100 1.0 100
Average 1.3 95 1.0 95
```

The variable `runq-sz` (number of threads in memory waiting execution) may be not greater than 4, other CPU has a bottleneck.

### 3.4. Monitoring server memory and client memory

For performance management, you must ensure that the device and network have sufficient memory, because when devices run out

of memory, some operations on the device can fail.

Command # netstat -m displays statistics for memory buffers, in case of many processors displays information of each processor.

```
# netstat -m
190/285 mbufs in use:
88 mbufs allocated to data
3 mbufs allocated to packet headers
23 mbufs allocated to socket structures
67 mbufs allocated to protocol control blocks
3 mbufs allocated to routing table entries
3 mbufs allocated to socket names and addresses
# netstat -i 5
      input      enl      output      input      (Total)output
      packets  errs  packets  errs  colls  packets  errs  packets  errs  colls
16          0      1          0      0      16          0      1          0      0
34          0      2          0      0      34          0      2          0      0
94          0      2          0      0      94          0      2          0      0
104         0      2          0      0      104         0      2          0      0
117         0      2          0      0      117         0      2          0      0
```

The netstat -v command displays the statistics for each interface. Every interface has its own specific information and some general information. With a different adapter, the statistics will differ somewhat. The most important output fields are highlighted.

**4. Calculus for collisions rate and network load**

Calculate the network collisions rate by dividing the number of packet collisions detected (field colls for output from command netstat -i) by the number of packets transmitted (field output packets for output from command netstat -i):

$$\text{collrate} = \frac{\text{colls}}{\text{Opkts}} * 100 \text{ [percent],}$$

For first example collrate=1828/30072 = 6.0787%

**Note:** If this percent is greater than 10% can indicate a problem, and the network must be reorganize, because there is high network utilization.

Calculate the Ethernet network overload using formula:

$$\text{load} = \frac{\text{coll} + \text{errs}}{\text{transmit packets}} * 100 \text{ [percent],}$$

where coll represents number of unsuccessful transmissions due to too many collisions and errs represents the number of unsuccessful transmissions due to adapter reported timeout

```
2 mbufs allocated to interface addresses
1 mbufs allocated to internet multicast options
.....
```

**3.5. Monitoring server network interface and client network interface**

Interfaces refer to the amount of data that can be flow simultaneously on any one connection.

Command netstat -i displays the activity of the default network interface, "en1", for 5 seconds test.

errors. Those are highlighted fields for command netstat -v.

If the result is greater than 5%, reorganize the network, using segmentation for load balancing.

**5. Conclusions**

This article presents how can the clients measuring network for many factors, which affect performance. An advantage measuring network performance by system commands method is that they require no participation of a network administrator. User can calculate the network elements affecting performance can manage the network. The same idea writes George Hamilton, director of enterprise computing and networking at the Yankee Group in the Network World in June 2006: "Today applications are much more complex and no longer tightly coupled to the underlying infrastructure".

**Bibliography**

1. V. Paxson, G.Almes, J.Mahdavi, M. Mathis – Framework fot IP performance metrics, RFC 2330, May 1998
- 2.Geoff Huston – Internet Performance Survival Guide, Wiley Computer Publishing, 2000
3. Trends in Measurement and Monitoring of Internet Backbones, session at the 26th North American Network Operators Group, hosted by D. Meyer, October 2002
4. Dave Olker - Network Considerations for Optimizing NFS Performance, Prentice Hall PTR, January 2003
5. Peer Hasselmeier - An Infrastructure for the Management of Dynamic Service Networks, IEEE Communications Magazine, vol. 41, no. 4, April 2003
6. Richard Blum - Network Performance Open Source Toolkit: Using Netperf, tcptrace, NISTnet, and SSFNet, Wiley Computer Publishing, July 2003
7. <http://www.freebsd.org/projects/netperf/index.html>  
The latest version of the source code netperf is available at the following location:  
<ftp://ftp.cup.hp.com/dist/networking/benchmarks/netperf>