TCP/IP Troubleshooting Tips & Tools

Gordon Webber
William Data Systems

August 2011
AGENDA

- Know Your Network
- Action Plans / Problem Determination
- Tools – General Usage
- Understanding the Common Tools (ping, traceroute, netstat, nslookup, ...)
- Problem Diagnosis Tips
• In order to manage any network successfully, you must be aware of the topology.

• Before any successful, and timely, problem resolution can be attempted, a (current!) network diagram is essential.

• The diagram (and associated documentation) should indicate all nodes and all possible paths, and detail the subnets, addresses and software (especially versions) available at each node.

• Only then is it possible to create an appropriate action plan...
• **Where to Start?** - First, *identify the problem*. This will determine the right tools to use, and the right place to start testing from (!”Top-down” or “Bottom-up”!). Progressive testing may be needed to isolate the problem area.

• Network problems usually fall into two or three categories:-
  
  • **No connection can be made.**
  
  • **Connections can be made, but are unstable, OR, not all functions operate.**
  
  • **Connections are stable but performance is poor.**
Action Plans . . . .

**Connectivity** issues can be caused by:-

- Application errors
- Failed network connections
- Bad configuration/changes
- Hardware failures
- Failed bind
- Power failures
- Security restrictions

**Performance** issues can be caused by:-

- Insufficient bandwidth
- Bottlenecks
- Priorities
- Retries
- Broadcasts
- Congestion
- Routing
- Fragmentation
- Application errors
- Switch faults
1. **Investigate (ALL) error messages** – these may indicate the nature and location of the failure [e.g. “ttl” expired, no path available, packet size too large (“nofragment” is on)].

2. **Classify the error** – ask what works and what doesn’t, and for whom . . .

- Problems affecting only one person may be local and physical (e.g. check the cables/switch/vlan first).
- Problems affecting more than one user are more likely to be the network.
- Problems affecting more than one person & more than one network path are more likely to be the application.

!! Syslogd!!
3. **Test connectivity** *(end-to-end)* – using Ping/Traceroute.
Be careful to ensure that the packets take the same path as the problem connection (i.e. ensure correct source interface address – you may need to use an “extended” PING).

- If PING fails, note the location and investigate there.
- If PING succeeds (note that this is ICMP, the connection probably uses TCP, so this may **NOT** be a conclusive test), try with a TCP PING if available
- If PING succeeds try again with larger packets, if appropriate.
For Example: Problem reported as ... "end-user cannot connect to application"

- Starting at the end-user system ensure local physical connections are good, then check the next layer, such as local switch ports, vlans, routers, and even firewalls.

- Then, test each “hop” by progressive steps across the network.

- Then ensure that the system running the required application is connected at the network level ("ping" from that system outbound via the interface in question.

If all these results are good, then the issue is probably with the application and not a network problem!
Disclaimer:

The fact that some tools are mentioned in this presentation while other tools are not, in no way implies recommendation of the tools mentioned, nor condemnation of those tools not mentioned.

The purpose of this presentation is simply to make attendees aware that such tools exist, and the attendees should make up their own mind as to the suitability of any tool used on their own system.
“Common” Tools . . .

“PING”
- proves that connectivity exists

“TRACERTE”
“tracert”
- discovers the network path (also

“NETSTAT”
- to locate connection information

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>All connections to a stack</td>
</tr>
<tr>
<td>ALLConn</td>
<td>TCP/IP connections</td>
</tr>
<tr>
<td>ARP</td>
<td>Query ARP table or entry information</td>
</tr>
<tr>
<td>CONFIG</td>
<td>Configuration data</td>
</tr>
<tr>
<td>Conn</td>
<td>Active TCP/IP connections (Default)</td>
</tr>
<tr>
<td>DEVlinks</td>
<td>Devices and links</td>
</tr>
<tr>
<td>Gate</td>
<td>Current known gateways</td>
</tr>
<tr>
<td>HOME</td>
<td>Home address list</td>
</tr>
<tr>
<td>PORTList</td>
<td>Display port reservation list</td>
</tr>
<tr>
<td>ROUTe</td>
<td>Display routing information</td>
</tr>
<tr>
<td>SOCKets</td>
<td>Socket interface users and sockets</td>
</tr>
<tr>
<td>STATS</td>
<td>TCP/IP statistics</td>
</tr>
<tr>
<td>TCP</td>
<td>Displays detailed info about the stack</td>
</tr>
<tr>
<td>TELnet</td>
<td>Telnet connection information</td>
</tr>
</tbody>
</table>

z/OS command format:

```
-------------------
NETSTAT < Option | Command > <
Target > < Output > < (Select >

E.g.:
TSO NETSTAT CONN (PORT 25
TSO NETSTAT TCP TCPIP
```

Note that “NETSTAT .....(REPORT” will collect the output
to a dataset; for ease of reading or input to a REXX?
Other Tools . . . .

“Nslookup” - test domain name resolution (& “DIG”)

“Snmp” - where SNMP is supported, there are many tools available to extract further information (MIB data), once the problem area has been located (e.g. Monitors, such as “Implex” for z/OS ; “iReasoning” elsewhere)

- - - - -

“TIVOLI” - IBM network tools (Monitor and trace facilities)

“Ctrace” - z/OS trace tool
“EXIGENCE” - WDS trace “expert” system
(now ZTS ! – “ZEN Trace & Solve”)
“TPing” - (“TurboPing”) “PING” using TCP packets
“Tcpdump” - (also Windump & SSLdump) is a packet sniffer found on many (most?) open platforms.
“Ethereal” - open system packet analyser (& “Wireshark”)
“Pchar” - is a reimplementation of Van Jacobson's (“Mr Traceroute”) `pathchar` utility which analyses the individual hops of a path.
“Netcat” - Netcat is a utility which reads and writes data across network connections. It is a network debugging and exploration tool. (+ `port-scanner` !)
“VisualRoute” - path checker and graphical display
“NeoTrace” - (McAfee) Internet locator: enhanced traceroute
....etc
“Ping” - “Packet INternetwork Groper”, is usually ICMP-based, which works if ICMP is allowed to pass. If not permitted, then an application-based ping can be used [e.g. “APING” (UDP) or “TPing” (TCP)].

Ping tests by sending out **ICMP Request** packets, and receiving **ICMP Replies**, therefore verifying up to (ISO) **layer 3** . . .

```
C:\>ping 66.249.85.99          ( www.google.co.uk ----- use IP address or URL )
Pinging 66.249.85.99 with 32 bytes of data:

Reply from 66.249.85.99: bytes=32 time=22ms TTL=244
Reply from 66.249.85.99: bytes=32 time=22ms TTL=244
Reply from 66.249.85.99: bytes=32 time=42ms TTL=244
Reply from 66.249.85.99: bytes=32 time=22ms TTL=244

Ping statistics for 66.249.85.99: Packets: Sent=4, Recvd=4, Lost=0 (0% loss), Approx. round trip times in milliseconds: Min=22ms, Max=42ms, Ave=27ms
```
ISO 7-Layer Network Model

Layer 1: Physical - defines the real hardware.
Layer 2: Data Link - defines the format of data (frame/packet). (MAC)
Layer 3: Network - responsible for routing datagrams. (IP)
Layer 4: Transport - manages data between network and user. (TCP/UDP)
Layer 5: Session - defines the format of the data sent.
Layer 6: Presentation - converts to/from local representation of data.
Layer 7: Application - provides network services to the end-users.

TCP/IP 4-Layer (Unix/DoD) Network Model

Layer 1: Link - defines the network hardware and device drivers.
Layer 2: Network - addressing, routing, delivery. (IP / ICMP) (ARP)
Layer 3: Transport - communication; end-to-end integrity. (TCP / UDP)
Layer 4: Application - user applications. (DNS, arp, telnet, smtp, http, ftp, traceroute,...)
### ICMP Types:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Echo Reply</td>
</tr>
<tr>
<td>3</td>
<td>Destination Unreachable</td>
</tr>
<tr>
<td>4</td>
<td>Source Quench</td>
</tr>
<tr>
<td>5</td>
<td>Redirect</td>
</tr>
<tr>
<td>6</td>
<td>Alternate Host Address</td>
</tr>
<tr>
<td>8</td>
<td>Echo</td>
</tr>
<tr>
<td>9</td>
<td>Router Advertisement</td>
</tr>
<tr>
<td>10</td>
<td>Router Solicitation</td>
</tr>
<tr>
<td>11</td>
<td>Time Exceeded</td>
</tr>
<tr>
<td>12</td>
<td>Parameter Problem</td>
</tr>
<tr>
<td>13</td>
<td>Timestamp</td>
</tr>
<tr>
<td>14</td>
<td>Timestamp Reply</td>
</tr>
<tr>
<td>15</td>
<td>Information Request</td>
</tr>
<tr>
<td>16</td>
<td>Information Reply</td>
</tr>
<tr>
<td>17</td>
<td>Address Mask Request</td>
</tr>
<tr>
<td>18</td>
<td>Address Mask Reply</td>
</tr>
<tr>
<td>30</td>
<td>Traceroute</td>
</tr>
<tr>
<td>31</td>
<td>Datagram Conversion Error</td>
</tr>
<tr>
<td>32</td>
<td>Mobile Host Redirect</td>
</tr>
<tr>
<td>33</td>
<td>IPv6 Where-Are-You</td>
</tr>
<tr>
<td>34</td>
<td>IPv6 I-Am-Here</td>
</tr>
<tr>
<td>35</td>
<td>Mobile Registration Request</td>
</tr>
<tr>
<td>36</td>
<td>Mobile Registration Reply</td>
</tr>
<tr>
<td>37</td>
<td>Domain Name Request</td>
</tr>
<tr>
<td>38</td>
<td>Domain Name Reply</td>
</tr>
</tbody>
</table>

### ICMP Codes:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Destination Unreachable</td>
</tr>
<tr>
<td>0</td>
<td>Net Unreachable</td>
</tr>
<tr>
<td>1</td>
<td>Host Unreachable</td>
</tr>
<tr>
<td>2</td>
<td>Protocol Unreachable</td>
</tr>
<tr>
<td>3</td>
<td>Port Unreachable</td>
</tr>
<tr>
<td>4</td>
<td>Fragmentation Needed and DF Set</td>
</tr>
<tr>
<td>5</td>
<td>Source Route Failed</td>
</tr>
<tr>
<td>6</td>
<td>Destination Network Unknown</td>
</tr>
<tr>
<td>7</td>
<td>Destination Host Unknown</td>
</tr>
<tr>
<td>8</td>
<td>Source Host Isolated</td>
</tr>
<tr>
<td>9</td>
<td>Communication with Dest Network Prohibited</td>
</tr>
<tr>
<td>10</td>
<td>Communication with Dest Host Prohibited</td>
</tr>
<tr>
<td>11</td>
<td>Dest Network Unreachable for Type of Service</td>
</tr>
<tr>
<td>12</td>
<td>Dest Host Unreachable for Type of Service</td>
</tr>
<tr>
<td>13</td>
<td>Communication Administratively Prohibited</td>
</tr>
<tr>
<td>14</td>
<td>Host Precedence Violation</td>
</tr>
<tr>
<td>15</td>
<td>Precedence cutoff in effect</td>
</tr>
<tr>
<td>11</td>
<td>Time Exceeded</td>
</tr>
<tr>
<td>0</td>
<td>Time to Live exceeded in Transit</td>
</tr>
<tr>
<td>1</td>
<td>Fragment Reassembly Time Exceeded</td>
</tr>
</tbody>
</table>

Ref: [“www.iana.org/assignments/icmp-parameters”](http://www.iana.org/assignments/icmp-parameters)

Options:
- **-t** Ping the specified host until stopped.
  To see statistics and continue - type Control-Break;
  To stop - type Control-C.
- **-a** Resolve addresses to hostnames.
- **-n** count Number of echo requests to send.
- **-l** size Send buffer size.
- **-f** Set Don't Fragment flag in packet.
- **-I** TTL Time To Live.
- **-v** TOS Type Of Service.
- **-r** count Record route for count hops.
- **-s** count Timestamp for count hops.
- **-j** host-list Loose source route along host-list.
- **-k** host-list Strict source route along host-list.
- **-w** timeout Timeout in milliseconds to wait for each reply.
C:\>ping 66.249.85.55 \non-existent addresses
Pinging 66.249.85.55 with 32 bytes of data:

Request timed out.
Request timed out.  (or “Destination Unreachable ?)
Request timed out.  (if a return path is available)
Request timed out.

Ping statistics for 66.249.85.55: Packets: Sent=4, Recvd=0, Lost=4 (100% loss),

**Drawbacks:**
- Extra traffic on the network.
- “**Time To Live**” (TTL) set to a high value to ensure penetration.
- Network devices **may not allow** Ping/ICMP and may drop its priority.
- May not take the same path as user traffic; delay (latency) reported may **not** be representative for the application(s).
- Low feedback on fault and location.
**Usage:** `tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] target_name

**Options:**
- `-d` Do not resolve addresses to hostnames.
- `-h maximum_hops` Maximum number of hops to search for target.
- `-j host-list` Loose source route along host-list.
- `-w timeout` Wait timeout milliseconds for each reply.

- Also uses ICMP! (although some platforms use UDP)
- Good for spotting “loops” in the routing
- “**Time To Live**” (TTL*) is incremented for each positive response.
- Each “hop” in the path is identified (Names may be resolved!).
- “Per hop” round-trip delays can be identified.
- **Drawbacks** are similar to those of “Ping”.

( * = anti-looping function of TCP/IP )
C:\>tracert 66.249.85.55 (www.google.co.uk) ----- use IP address or URL

Tracing route to 66.249.85.55 over a maximum of 30 hops

1 1 ms 1 ms 1 ms 81.144.212.33
2 7 ms 6 ms 6 ms 62.7.96.41
3 6 ms 6 ms 6 ms core2-gig2-1.kingston.ukcore.bt.net [194.72.3.2]
4 7 ms 7 ms 7 ms core2-pos7-3.ealing.ukcore.bt.net [62.6.201.42]
5 7 ms 7 ms 7 ms core2-pos10-0.redbus.ukcore.bt.net [194.74.65.202]
6 8 ms 7 ms 8 ms 194.74.65.38
7 7 ms 7 ms 7 ms 72.14.238.244
8 16 ms 16 ms 16 ms 216.239.43.91
9 22 ms 22 ms 22 ms 72.14.232.209
10 * * * Request timed out.
11 * * * Request timed out.
12 * etc, etc . . . ------ default maximum of 30

TRACEROUTE should be run in BOTH directions!!

Look for unsuitable (long) routes and high latency
Some platforms give status indicators...

!H - Host unreachable. (Destination Net unreachable) The router has no route to the target system.

!N - Network unreachable.

!P - Protocol unreachable.

!S - Source route failed. A router is blocking source-routed packets.

!F - Fragmentation needed. (Check the MTU configuration at the router).

!X - Communication administratively prohibited. Traceroute blocked!

TRACEROUTE can be enhanced by visualization, as is often seen in graphical traceroute tools: such as ...
VisualRoute - 1

Report for www.google.co.uk [66.249.85.99]

Analysis:
This trace was started on 09-Jan-2007 19:38:48. The host 'www.google.co.uk' (known as P-in-199.google.com) has been found, and is reachable in 10 hops. Also, it responded to HTTP requests on port 80 (it is running server GWS/2.1, which responded in 431ms). The TTL value of packets received from it is 245.

In general this route offers a good throughput, with hops responding on average within 11ms. The DNS lookup was completed almost instantaneously (less than 2ms - this may be the result of caching).
VisualRoute - 2

TraceRoute Tools

Learn more at:
http://www.visualroute.com
## TraceRoute Tools

**Target Name:** www.google.co.uk  
**IP:** 66.249.85.99

<table>
<thead>
<tr>
<th>Hop</th>
<th>PL%</th>
<th>IP</th>
<th>DNSName</th>
<th>Avg</th>
<th>Cur</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81.144.212.33</td>
<td>-------</td>
<td>------</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>62.7.96.41</td>
<td>-------</td>
<td>------</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>194.72.3.2</td>
<td>core2-gig2-1.kingston.uk.core.bt.net</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>62.6.201.42</td>
<td>core2-pos7-3.ealing.uk.core.bt.net</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>194.74.65.202</td>
<td>core2-pos10-0.redbus.uk.core.bt.net</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>194.74.65.38</td>
<td>------</td>
<td>------</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>72.14.238.244</td>
<td>------</td>
<td>------</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>216.239.43.91</td>
<td>------</td>
<td>------</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>72.14.232.209</td>
<td>------</td>
<td>------</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>66.249.85.99</td>
<td>if-in-f99.google.com</td>
<td>21</td>
<td>22</td>
<td></td>
</tr>
</tbody>
</table>

**Round Trip:** 21 22

Data and Image generated by Ping Plotter Freeware (http://www.pingplotter.com)
Where the target system is external to the local network, and especially where routing is not available to/from the local network, there are several sites around the World that offer the ability to run “Ping” and “Traceroute” to be instigated by remote control from their web site.

Basically, this is a “proxy” service; the remote site issuing the test on your behalf.

This is suitable for determining the general availability of the target system (i.e. from anywhere on the Internet), but does not test specific routes.

“www.samspade.org” used to be an excellent example of this type of service, but is not currently available in its previous form.

Further directions to such services can be found at :-

“www.traceroute.org”
NETSTAT (z/OS)

NETSTAT < Option | Command > < Target >
< Output > < (Select >

TSO NETSTAT CONN      TSO NETSTAT SOCK
TSO NETSTAT DEV        TSO NETSTAT ROUTE
TSO NETSTAT TCP TCPIP

Also “onetstat”...

Can be issued from either TSO or USS; the results are the same.

NB. Netstat options will vary depending upon the platform!

Note the following examples from z/OS and Windows. . .
**Tools in Detail . . . . .**

**NETSTAT(z/OS) –”DEV”**

**DevName:** LCS1  
**DevType:** LCS  
**DevNum:** 0E20  
**DevStatus:** Ready  

**LnkName:** ETH1  
**LnkType:** ETH  
**LnkStatus:** Ready  

- **NetNum:** 3  
- **QueSize:** 0  
- **IpBroadcastCapability:** Yes  
- **MacAddress:** 000255305115  
- **ActMtu:** 1500  

**BSD Routing Parameters:**  
- **MTU Size:** 00000  
- **Metric:** 00  
- **DestAddr:** 0.0.0.0  
- **SubnetMask:** 255.255.0.0  

**Packet Trace Setting:**  
- **Protocol:** 253  
- **SrcPort:** *  
- **IpAddr:** *  

**Multicast Specific:**  
- **Multicast Capability:** Yes  
- **Group**  
- **RefCnt**  

- **224.0.0.1**  
  **RefCnt:** 0000000001  

**Link Statistics:**  
- **BytesIn:** 420328206  
- **Inbound Packets:** 2865741  
- **Inbound Packets In Error:** 1360  
- **Inbound Packets Discarded:** 0  
- **Inbound Packets With No Protocol:** 0  

**NETSTAT(z/OS) –”SOCK”**

**MVS TCP/IP NETSTAT CS V1R5**  
**TCPIP Name:** TCPIP  

**Name:** APIASHB  
**Subtask:** 007E1048  
**Type:** Dgram  
**Status:** UDP  
**Conn:** 00001A1A  
**BoundTo:** 192.168.1.156..12004  
**ConnTo:** *.*.*  

**Type:** Stream  
**Status:** Listen  
**Conn:** 00001A19  
**BoundTo:** 192.168.1.156..12004  
**ConnTo:** 0.0.0.0..0  

**Name:** APIASHB  
**Subtask:** 007E12D8  
**Type:** Dgram  
**Status:** UDP  
**Conn:** 00001A18  
**BoundTo:** 192.168.1.156..12000  
**ConnTo:** *.*.*  

**Type:** Stream  
**Status:** Listen  
**Conn:** 00001A17  
**BoundTo:** 192.168.1.156..12000  
**ConnTo:** 0.0.0.0..0
Tools in Detail

**NETSTAT (Windows)**


- `-a` Displays all connections and listening ports.
- `-n` Displays addresses and port numbers in numerical form.
- `-r` Displays the routing table.

...etc

C:\>netstat -a

<table>
<thead>
<tr>
<th>Proto</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>wdsgdw:epmap</td>
<td>0.0.0.0:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wdsgdw:microsoft-ds</td>
<td>0.0.0.0:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wdsgdw:1028</td>
<td>0.0.0.0:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wdsgdw:1241</td>
<td>0.0.0.0:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wdsgdw:10110</td>
<td>0.0.0.0:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>UDP</td>
<td>wdsgdw:microsoft-ds</td>
<td><em>:</em></td>
<td></td>
</tr>
<tr>
<td>UDP</td>
<td>wdsgdw:isakmp</td>
<td><em>:</em></td>
<td></td>
</tr>
<tr>
<td>UDP</td>
<td>wdsgdw:1033</td>
<td><em>:</em></td>
<td></td>
</tr>
<tr>
<td>UDP</td>
<td>wdsgdw:4500</td>
<td><em>:</em></td>
<td></td>
</tr>
<tr>
<td>UDP</td>
<td>wdsgdw:ntp</td>
<td><em>:</em></td>
<td></td>
</tr>
<tr>
<td>UDP</td>
<td>wdsgdw:1900</td>
<td><em>:</em></td>
<td></td>
</tr>
</tbody>
</table>
In general, it is quite common to seek an IP target using a URL (which acts rather like a PATH name).

This entails sending the URL to a “Domain Name Server” (or “Resolver”) in z/OS terms) to have the name translated (i.e. a “table lookup”) into an IP address (this may occur locally by use of the “Hosts” file).

The IP address returned is then used to address the target.

This process may also be performed in reverse; i.e. the DNS server can translate an IP address into a URL!

The use of a URL means that remote services can be failed-over, relocated or rebuilt without the users needing to know!
The global Domain Name System is a hierarchy of servers/services spread across the Internet. At its core is a set of servers that manage the base domains; such as “com”, “edu”, “gov” ...etc

When a name is “looked up” it happens from right to left - recursively.

Take www.google.co.uk ...

. First the server is located that controls the “uk” domain (there is an implied “root” service where all top-level servers are known).
. This will indicate the “co.uk” server; which in turn will indicate the “google.co.uk” server.
. The “google.co.uk” server will have IP addresses (an “A” record) for web (“www”) and mail services (note: “www” is not the only canonical form used!)

NAMED.CONF - lists the “zones” (eg. “google.co.uk”)
ZONE FILES - hold the IP addresses

NB. Zone information changed at the bottom of a “layer” is propagated upwards by “Zone Transfer” at preset times.
**NSLOOKUP (Windows)**

**Usage:**
- `nslookup NAME`, or, `NAME1 NAME2`
- `command`

**set option**
- `[no]recurse` [no]search [no]vc domain=NAME
- `srchlist=N1[/N2/.../N6]` root=NAME retry=x
- `timeout=X` type=X querytype=X class=X
- `[no]msxfr` ixfrver=X

**Server NAME**

**Exit**

“Lookup” failure will cause connectivity failure, and symptoms can be mistaken for a routing problem!

---

z/OS often acts as a relay, passing the requests on to a network DNS server.
C:\>nslookup

> set debug
> www.google.co.uk
Server:  my.router
Address:  192.168.27.1
---------- (debug information)
Got answer:
   HEADER:
      opcode = QUERY, id = 3, rcode = NOERROR
      header flags:  response, want recursion, recursion avail.
      questions = 1,  answers = 1,  authority records = 0,  additional = 0

   QUESTIONS:
      www.google.co.uk.uk.willdata.com, type = A, class = IN

   ANSWERS:
      -> www.google.co.uk.uk.willdata.com
         internet address = 212.69.199.183
         ttl = 60 (1 min)

----------
Non-authoritative answer:             ------( Retrieved from a cache! )
Name:   www.google.co.uk.uk.willdata.com
Address:  212.69.199.183
Domain Internet Groper: A tool for system administrators; it issues DNS queries and formats/interprets the answers.... Quite popular (allegedly!) with hackers...

Usage: dig [@global-server] [domain] [q-type] [q-class] {q-opt} {global-d-opt} host [@local-server] {local-d-opt} [ host [@local-server] {local-d-opt} [...]]

dig @lizzie www.google.co.uk any
; <<< DiG 9.3.1 <<< @lizzie www.google.co.uk any
; (1 server found) ; global options: printcmd ; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16774
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;www.google.co.uk.              IN      ANY
;; ANSWER SECTION:
www.google.co.uk.       86399   IN      CNAME   www.google.com.
;; Query time: 63 msec
;; SERVER: 192.168.1.45#53(192.168.1.45)
;; WHEN: Mon Feb 5 14:11:43 2007
;; MSG SIZE  rcvd: 62
dig @lizzie www.google.com any

; <<>> DiG 9.3.1 <<>> @lizzie www.google.com any
; (1 server found) ; global options: printcmd ; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60773
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 3

;; QUESTION SECTION:
;www.google.com.                        IN      ANY

;; ANSWER SECTION:

;; ADDITIONAL SECTION:
www.l.google.com.       149     IN      A       66.249.93.104
www.l.google.com.       149     IN      A       66.249.93.99
www.l.google.com.       149     IN      A       66.249.93.147

;; Query time: 56 msec
;; SERVER: 192.168.1.45#53(192.168.1.45)
;; WHEN: Mon Feb 5 14:15:13 2007
;; MSG SIZE   rcvd: 100
Domain name: google.co.uk

Registrant:
Google Inc

Registrant type:
Non-UK Corporation

Registrant's address:
1600 Amphitheatre Parkway
Mountain View
CA
94043
United States

Registrant's agent:
Markmonitor Inc. t/a Markmonitor
URL: http://www.markmonitor.com

Relevant dates:
Registered on: 14-Feb-1999
Renewal date: 14-Feb-2009
Last updated: 17-Jan-2007

Registration status:
Renewal request being processed.

Name servers:
ns1.google.com
ns2.google.com
ns3.google.com
ns4.google.com
**Pchar**

Estimates bandwidth, latency and packet loss on network links.

This is a re-working of the “pathchar” utility, written by Van Jacobson and, like traceroute, is based on repeated packet transmission and TTL variation (it can use ICMP or UDP).

It is available for most “*nix” systems: It works for IPv4 & IPv6.

Traceroute (UDP) knows when it has found its target by using a port number beyond the “normal range”… when ICMP “port unreachable” is returned it’s there!

Pchar sends many packets, one hop at a time, with varying the sizes, until the target is reached or the path fails. It calculates the latency from the ICMP message response times, and the throughput per hop from the variance in response speeds. Collectively, this also gives the overall round-trip delay for the whole path.

It is not fool-proof; it’s traffic may not be allowed; it is not a “Holy Grail”; but it does give a good indication!
pchar to www.l.google.com (66.249.93.104) using UDP/IPv4
Using raw socket input
Packet size increments from 32 to 1500 by 32
46 test(s) per repetition : 32 repetition(s) per hop
Warning: target host did not respond to initial test.
0: 192.168.1.231 (dhcp-192-168-1-231.uk.willdata.com)
  Partial loss: 0 / 1472 (0%)
  Partial char: rtt = 0.959029 ms, (b = 0.001150 ms/B), r2 = 0.999475
  Partial queueing: avg = 0.000171 ms (148 bytes)
  Hop char: rtt = 0.959029 ms, bw = 6954.330709 Kbps
  Hop queueing: avg = 0.000171 ms (148 bytes)
1: 81.144.212.33 (81.144.212.33)
  Partial loss: 0 / 1472 (0%)
  Partial char: rtt = 5.784087 ms, (b = 0.005317 ms/B), r2 = 0.999798
  Partial queueing: avg = 0.002336 ms (667 bytes)
  Hop char: rtt = 4.825058 ms, bw = 1919.855256 Kbps
  Hop queueing: avg = 0.002165 ms (519 bytes)
2: 62.7.96.41 (62.7.96.41)
  Partial loss: 0 / 1472 (0%)
  Partial char: rtt = 5.824306 ms, (b = 0.005317 ms/B), r2 = 0.999847
  Partial queueing: avg = 0.001486 ms (667 bytes)
  Hop char: rtt = 0.040220 ms, bw = --.--- Kbps
  Hop queueing: avg = -0.000850 ms (0 bytes)
3: 194.72.3.66 (core2-gig10-1.kingston.ukcore.bt.net)
  process hangs at this point!

This example shows a “pchar” test across a path where icmp responses are not allowed.
pchar to 192.168.1.8 (192.168.1.8) using UDP/IPv4
Using raw socket input
Packet size increments from 32 to 1500 by 32
46 test(s) per repetition : 32 repetition(s) per hop

0: 192.168.1.231 (dhcp-192-168-1-231.uk.willdata.com)
    Partial loss: 0 / 1472 (0%)
    Partial char: rtt = 10.792415 ms, (b = 0.003369 ms/B), r2 = 0.157013
        stddev rtt = 0.950840, stddev b = 0.001177
    Partial queueing: avg = 0.015037 ms (4463 bytes)
    Hop char: rtt = 10.792415 ms, bw = 2374.706954 Kbps
    Hop queueing: avg = 0.015037 ms (4463 bytes)

1: 192.168.1.8 (zplex.uk.willdata.com)
    Partial loss = number of pkts / percentage pkts lost
    Partial char = RTT, delay Byte, min delay pkt
    Partial queueing = ave. queue of data incl. of this hop
    Hop char = RTT and b/width for the current hop
    Hop queueing = average queue of data this hop
    Path length: 1 hops
    Path char: rtt = 10.792415 ms r2 = 0.157013
    Path bottleneck: 2374.706954 Kbps
    Path pipe: 3203 bytes
    Path queueing: average = 0.015037 ms (4463 bytes)
    Start time: Thu Feb 1 09:07:32 2007
    End time: Thu Feb 1 09:14:22 2007
Remember:

ICMP may be restricted over the test path

Not all platforms have the same controls or defaults

Think of the impact on the network of using these kind of tools!!

The figures produced are estimates (ref. pchar “man pages” of pchar and, as already mentioned for some previous tools, the results will probably not reflect the exact behaviour of the applications using the same path.

Learn more at:
http://www.kitchenlab.org/www/bmah/Software/pchar/
Netcat - a read/write utility for networks (TCP or UDP). It can be used on its own or be driven by user code. It is also a very powerful network debugging and exploration tool, which can create almost any kind of connection:-

- Outbound or inbound, TCP or UDP, to or from any ports
- Full DNS forward/reverse checking, with appropriate warnings
- Ability to use any local source port
- Ability to use any locally-configured network source address
- Built-in port-scanning capabilities, with randomizer
- Can read command line arguments from standard input
- Slow-send mode, one line every N seconds
- Hex dump of transmitted and received data
- Ability to let another program service established connections
- Telnet-options responder

Good for testing applications and application paths, but does not “test” or measure the network itself.

Beware of misuse!
Netcat

connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [options] [hostname] [port]
options:
   -d              detach from console, background mode
   -e prog         inbound program to exec [dangerous!!]
   -g gateway      source-routing hop point[s], up to 8
   -G num          source-routing pointer: 4, 8, 12, ...
   -h              this help
   -i secs         delay interval for lines sent, ports scanned
   -l              listen mode, for inbound connects
   -L              listen harder, re-listen on socket close
   -n              numeric-only IP addresses, no DNS
   -o file         hex dump of traffic
   -p port         local port number
   -r              randomize local and remote ports
   -s addr         local source address
   -t              answer TELNET negotiation
   -u              UDP mode
   -v              verbose [use twice to be more verbose]
   -w secs         timeout for connects and final net reads
   -z              zero-I/O mode

port numbers can be individual or ranges: m-n [inclusive]

Learn more at:
http://netcat.sourceforge.net/
http://nmap.org/ncat/
Tools in Detail

**Netcat** - Retrieve page from web server

```
C:\>nc -v www.google.co.uk 80
www.l.google.com [216.239.59.103] 80 (http) open
GET / HTTP/1.0

HTTP/1.0 302 Found
Location: http://www.google.co.uk/
Cache-Control: private
Set-Cookie:
PREF=ID=bebf53d3e8c044c6:TM=1170500572:LM=1170500572:S=DBx029wrXh5ex5E;
expires=Sun, 17-Jan-2038 19:14:07 GMT
MT; path=/; domain=.google.com
Content-Type: text/html
Server: GWS/2.1
Content-Length: 221
Date: Sat, 03 Feb 2007 11:02:52 GMT
Connection: Keep-Alive

<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.co.uk/">here</A>.
</BODY></HTML>
```
**Netcat - “NC” to “NC” connection**

c:\> nc -l -p 23 -t -e cmd.exe

C:\Documents and Settings\gdw> netstat -a

<table>
<thead>
<tr>
<th>Proto</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>wds-gdw:ftp</td>
<td>wds-gdw.wds.local:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wds-gdw:telnet</td>
<td>wds-gdw.wds.local:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wds-gdw:epmap</td>
<td>wds-gdw.wds.local:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wds-gdw:microsoft-ds</td>
<td>wds-gdw.wds.local:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wds-gdw:1032</td>
<td>wds-gdw.wds.local:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wds-gdw:5354</td>
<td>wds-gdw.wds.local:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>wds-gdw:10110</td>
<td>wds-gdw.wds.local:0</td>
<td>LISTENING</td>
</tr>
</tbody>
</table>

192.168.27.10

C:\> nc 192.168.27.10 23

Microsoft Windows XP [Version 5.1.2600] ...

C:\> ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . :

IP Address . . . . . . . . . . . . : 192.168.27.10
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . : 192.168.27.1

C:\> ^C

C:\> ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . :

IP Address . . . . . . . . . . . . : 192.168.27.50
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . : 192.168.27.1
## SNMP - MIBs

Learn more at: [http://www.ireasoning.com/](http://www.ireasoning.com/)

### iReasoning MIB Browser

**Address:** 192.168.1.231:161

### SNMP MIBs

#### System
- **sysDescr**
- **sysObjectID**
- **sysUpTime**
- **sysContact**
- **sysName**
- **sysLocation**
- **sysServices**

#### Interfaces
- **iFNumber**
- **iFTable**
- **iFEntry**

#### FTable
- **FIndex**
- **FDescr**
- **FType**
- **FSpeed**
- **FPhysAddress**

### Interfaces Table

<table>
<thead>
<tr>
<th>Name/OID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.6.1.2.1.9.1.4.7</td>
<td>3</td>
</tr>
<tr>
<td>1.3.6.1.2.1.9.1.4.8</td>
<td>3</td>
</tr>
<tr>
<td>1.3.6.1.2.1.9.1.4.9</td>
<td>3</td>
</tr>
<tr>
<td>FIndex.1</td>
<td>1</td>
</tr>
<tr>
<td>FIndex.2</td>
<td>2</td>
</tr>
<tr>
<td>FIndex.3</td>
<td>3</td>
</tr>
<tr>
<td>FDescr.1</td>
<td>io</td>
</tr>
<tr>
<td>FDescr.2</td>
<td>eth0</td>
</tr>
<tr>
<td>FDescr.3</td>
<td>eth0</td>
</tr>
<tr>
<td>FType.1</td>
<td>software.v2i</td>
</tr>
<tr>
<td>FType.2</td>
<td>ethernet-canocd</td>
</tr>
<tr>
<td>FSpeed.1</td>
<td>131</td>
</tr>
<tr>
<td>FPhysicalAdd.1</td>
<td>Ox00 0x06 0x5E 0x37 oxP3 0x46</td>
</tr>
<tr>
<td>FAdminStatus.1</td>
<td>up</td>
</tr>
<tr>
<td>FOperStatus.1</td>
<td>up</td>
</tr>
<tr>
<td>FOperStatus.2</td>
<td>up</td>
</tr>
<tr>
<td>FOperStatus.3</td>
<td>down</td>
</tr>
</tbody>
</table>

### Learn more at:

<table>
<thead>
<tr>
<th>Object</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td></td>
</tr>
<tr>
<td>interfaces</td>
<td></td>
</tr>
<tr>
<td>ifNumber</td>
<td>3</td>
</tr>
<tr>
<td>ifTable</td>
<td></td>
</tr>
<tr>
<td>ifEntry</td>
<td></td>
</tr>
<tr>
<td>ifIndex</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>ifDescr</td>
<td>(I)</td>
</tr>
<tr>
<td>ifType</td>
<td>(I)</td>
</tr>
<tr>
<td>ifMtu</td>
<td>(I)</td>
</tr>
<tr>
<td>ifSpeed</td>
<td>(I)</td>
</tr>
<tr>
<td>ifPhysAddress</td>
<td>(I)</td>
</tr>
<tr>
<td>ifAdminStatus</td>
<td>(I)</td>
</tr>
<tr>
<td>ifOperStatus</td>
<td>(I)</td>
</tr>
<tr>
<td>ifLastChange</td>
<td>(I)</td>
</tr>
<tr>
<td>ifInOctets</td>
<td>(I)</td>
</tr>
<tr>
<td>ifInUcastPkts</td>
<td>(I)</td>
</tr>
<tr>
<td>ifInNUcastPkts</td>
<td>(I)</td>
</tr>
<tr>
<td>ifInDiscards</td>
<td>(I)</td>
</tr>
<tr>
<td>ifInErrors</td>
<td>(I)</td>
</tr>
<tr>
<td>ifInUnknownProtos</td>
<td>(I)</td>
</tr>
<tr>
<td>ifOutOfRange</td>
<td>(I)</td>
</tr>
<tr>
<td>ifOutOctets</td>
<td>(I)</td>
</tr>
<tr>
<td>ifOutUcastPkts</td>
<td>(I)</td>
</tr>
<tr>
<td>ifOutNUcastPkts</td>
<td>(I)</td>
</tr>
<tr>
<td>ifOutDiscards</td>
<td>(I)</td>
</tr>
<tr>
<td>ifOutErrors</td>
<td>(I)</td>
</tr>
<tr>
<td>ifOutQLen</td>
<td>(I)</td>
</tr>
<tr>
<td>ifSpecific</td>
<td>(I)</td>
</tr>
</tbody>
</table>

Objects: 265

7671
Tools in Detail . . . .

Packet Analysers – “Sniffers”

- “Original” capture routine - **TCPDUMP**  
  + **LIBPCAP** (the Promiscuous Capture Library) or **WinPcap**.  
  Available on most "open" platforms.

- **SSLDUMP** is TCPDUMP with SSL decryption capability.

- **ETHEREAL** is a packet analyzer based on TCPDUMP.

- **WIRESHARK** is the latest incarnation of ETHEREAL  
  Shows actual packets on the network with “breakdown”.  
  Good for true analysis of the network *and* for establishing  
  "common use" baselines.

- **EXIGENCE** provides similar functionality for z/OS.
The three panes show the traffic flow, the headers, and the data in dump format.

Highlighting is reflected in the lower panes.

This image shows the IP header...
This image shows the UDP header...
This image shows the DATA; in this case a DNS Query.
(http://www.wireshark.org/)
This image shows the equivalent displays in EXIGENCE; in this case for an FTP session. (http://www.willdata.com/)
Tools in Detail . . . . . .

“ZEN Trace and Solve”
Tools in Detail . . .

“ZEN Trace and Solve”

ZTS - Exigence in the ZEN Framework. (http://www.willdata.com/)
Network & Security testers

“Nessus” - (“The Tenable Newt”) a security vulnerability scanner.
(www.nessus.org)

“Nmap” - a network and security scanner
(insecure.org & nmap.org)

Use responsibly – Use with care!
```plaintext
Nmap (edited)

> nmap -v -A 192.168.27.50
Starting Nmap 4.20 (http://insecure.org) at 2007-02-03 11:40 GMT Standard Time
Initiating ARP Ping Scan at 11:40
Scanning 192.168.27.50 [1 port]
Completed ARP Ping Scan at 11:40, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:40
Completed Parallel DNS resolution of 1 host. at 11:40, 0.03s elapsed
Initiating SYN Stealth Scan at 11:40 : Scanning 192.168.27.50 [1697 ports]
Discovered open port 135/tcp on 192.168.27.50
Completed SYN Stealth Scan at 11:40, 39.05s elapsed (1697 total ports)
Initiating Service scan at 11:40 : Scanning 1 service on 192.168.27.50
Completed Service scan at 11:41, 11.63s elapsed (1 service on 1 host)
Warning: OS detection for 192.168.27.50 will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
...
Host 192.168.27.50 appears to be up ... good.
Interesting ports on 192.168.27.50:
Not shown: 1696 filtered ports
PORT      STATE SERVICE VERSION
135/tcp open  msrpc   Microsoft Windows RPC
MAC Address: xx:xx:xx:xx:xx:xx (Dell ESG Pcb Test)
Running (JUST GUESSING) : Microsoft Windows 2000|XP (98%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop : TCP Sequence Prediction: Difficulty=0 (Trivial joke)
...
OS and Service detection performed. Nmap finished: 1 IP address (1 host up) scanned in 67.000 seconds
Raw packets sent: 3517 (162.066KB) | Rcvd: 86 (4770B)
```

(NB. This sample has been edited to fit !)
Problem Diagnosis . . .

Outline Steps:

• Check the stack – “ping” local loopback
• “ping” the remote host/server name
• “ping” with IPaddress – the DNS may be down
• If “ping” fails “traceroute” - find where it stops
• Use “netstat” to check the interface
• Check routing (is it as expected?)
• If ping works, try “telnet” (standard port 23)
• If “telnet” works try telnet to the application port
• If that works try the application
• Use “netstat” to check the connection exists
• Check your syslogs (remember USS ! “syslogd” !)
• Do you still have a failure? ... trace it!
Summary . . . . .

• Know Your Network!

• Keep Up-to-Date Documentations & Diagrams!

• Know the Tools (most tools can be used for practice at any time)

• Plan Your Approach to Any Problem

• Stop, Look, and LISTEN!!
Thank you !