

S H A R E
Technology • Connections • Results

TCP/IP Troubleshooting Tips & Tools

Gordon Webber
William Data Systems

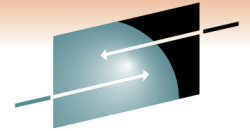
Session 8182
February 2011

Gordon.Webber@willdata.com

SHARE
in Anaheim
2011

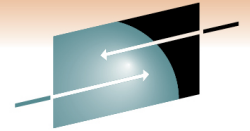


AGENDA



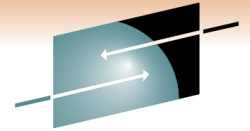
- **Know Your Network**
- **Action Plans / Problem Determination**
- **Tools – General Usage**
- **Understanding the Common Tools**
(ping, traceroute, netstat, nslookup, ...)
- **Problem Diagnosis Tips**

Know Your Network! . . .



- 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**...*

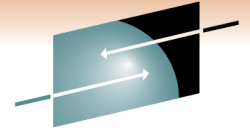
Action Plans



- **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.

Misinformation Anecdote

- 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.**



Connectivity issues can be caused by:-

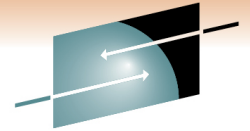
- Application errors
- Failed network connections
- Bad configuration/changes
- Hardware failures

- 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

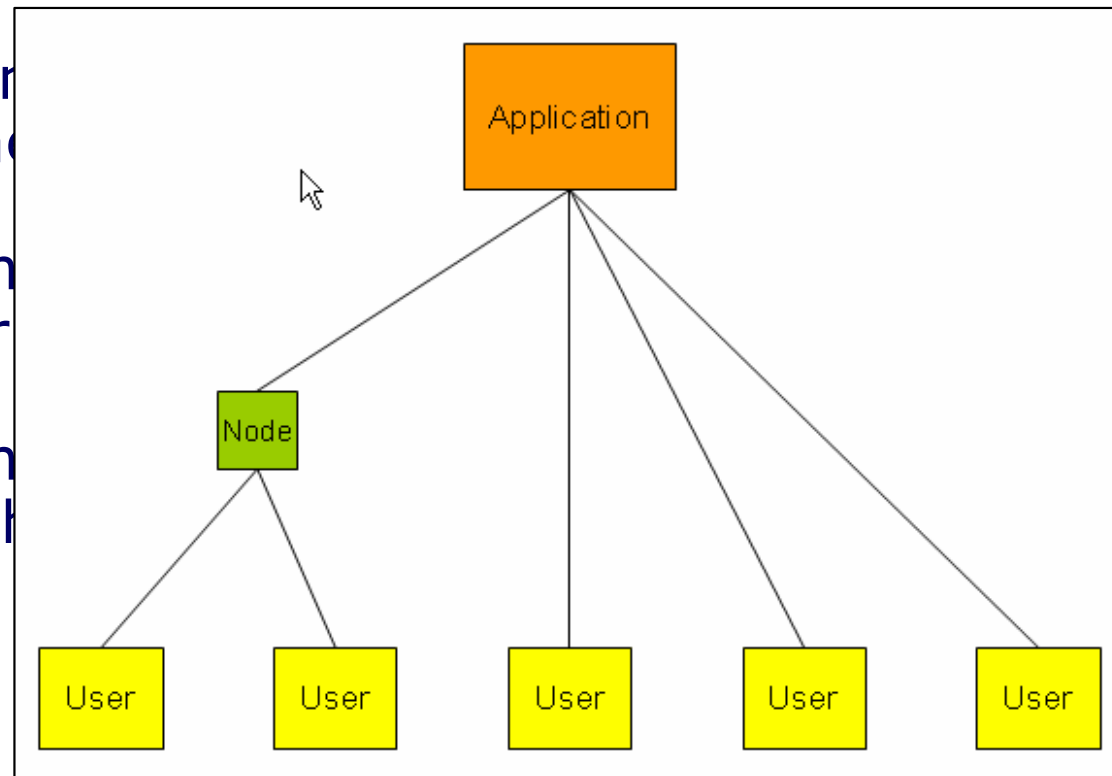


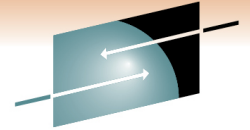
1. **Investigate (*ALL*) error messages** – these may indicate the nature and location of the failure [**e.g.** “ttl” expired, no path available, packet size too large (“nofragment” is on)].

!! Syslogd !!

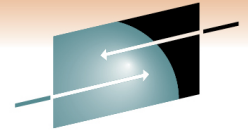
2. **Classify the error** – ask what works and what doesn't, and for whom . . .

- Problems affecting one or more physical (e.g. ch...
- Problems affecting multiple nodes to be the network...
- Problems affecting multiple nodes on one network path...





- 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.

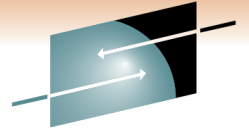


For Example: Problem reported as ...

“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”

- discovers the network path (also “tracert”)

“NETSTAT”

- to locate connection information

ALL	- All connections to a stack
ALLConn	- TCP/IP connections
ARp	- Query ARP table or entry information
CONFIG	- Configuration data
COnn	- Active TCP/IP connections (Default)
DEVlinks	- Devices and links
Gate	- Current known gateways
HOme	- Home address list
PORTList	- Display port reservation list
ROUTE	- Display routing information
SOCKETs	- Socket interface users and sockets
STATS	- TCP/IP statistics
TCP	- Displays detailed info about the stack
TELnet	- Telnet connection information

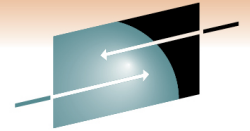
z/OS command format:

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)

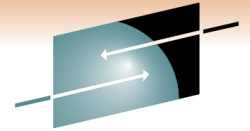
- - - - -

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

“Ctrace” - z/OS trace tool

“EXIGENCE” - WDS trace “expert” system
(now ZTS ! – “ZEN Trace & Solve”)

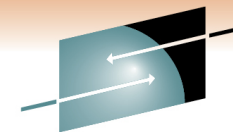
Other Tools



- "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 reimplementaion 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

* New *
Ncat from Nmap

....etc



“Ping”

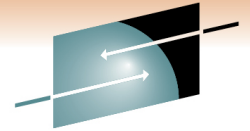
- “**P**acket **I**nternetwork **G**roper”, is usually ICMP-based, which works if ICMP is allowed to pass. If not permitted, then an application-based ping can be used [e.g. “**APING**” (UDP) or “TPing” (TCP)].

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
```

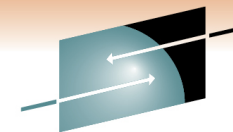


ISO 7-Layer Network Model

- 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....)

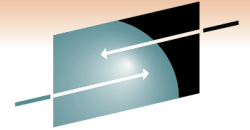


ICMP Types:

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

ICMP Codes:

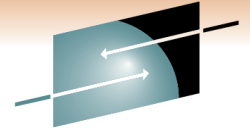
- 3** **Destination Unreachable**
 - 0 Net Unreachable
 - 1 Host Unreachable
 - 2 Protocol Unreachable
 - 3 Port Unreachable
 - 4 Fragmentation Needed and DF Set
 - 5 Source Route Failed
 - 6 Destination Network Unknown
 - 7 Destination Host Unknown
 - 8 Source Host Isolated
 - 9 Communication with Dest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

**PING** (Windows)

Usage: **ping** [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
[-r count] [-s count] [[-j host-list] | [-k host-list]]
[-w timeout] target_name

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. |



PING

```
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 ?)
```

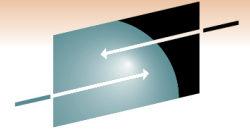
```
Request timed out. (if a return path is available)
```

```
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.



TRACEROUTE (Windows)

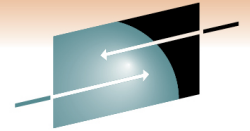
Usage: **tracert** [-d] [-h maximum_hops] [-j host-list]
[-w timeout] target_name

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*)



TRACEROUTE

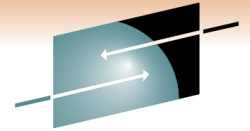
C:\>tracert 66.249.85.55 (www.google.co.uk ----- use *IP address or URL*)

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



TRACEROUTE

Some platforms give status indicators...

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

!N - Network unreachable.

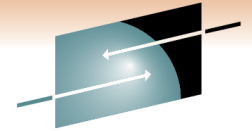
!P - Protocol unreachable.

!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]

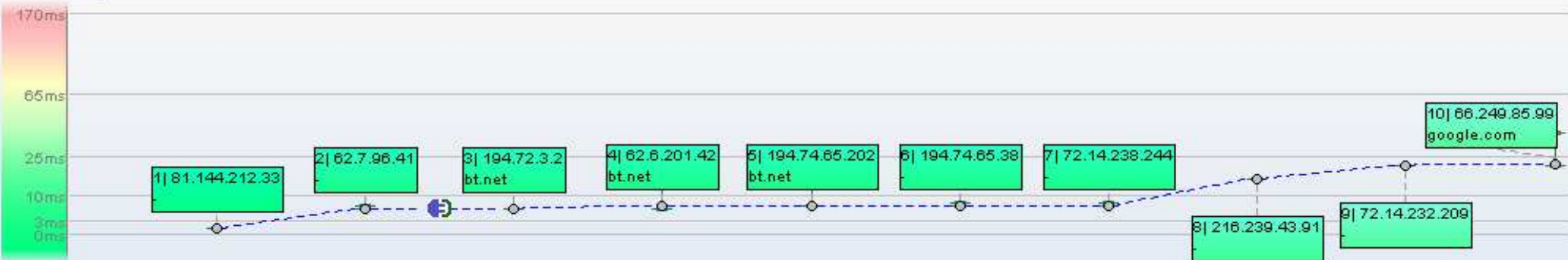
Analysis

This trace was started on 08-Jan-2007 10:28:48. The host 'www.google.co.uk' (known as ff-in-f99.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 431 ms). The [TTL value](#) of packets received from it is 246. 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).

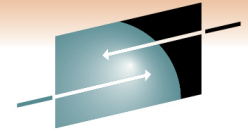
Map



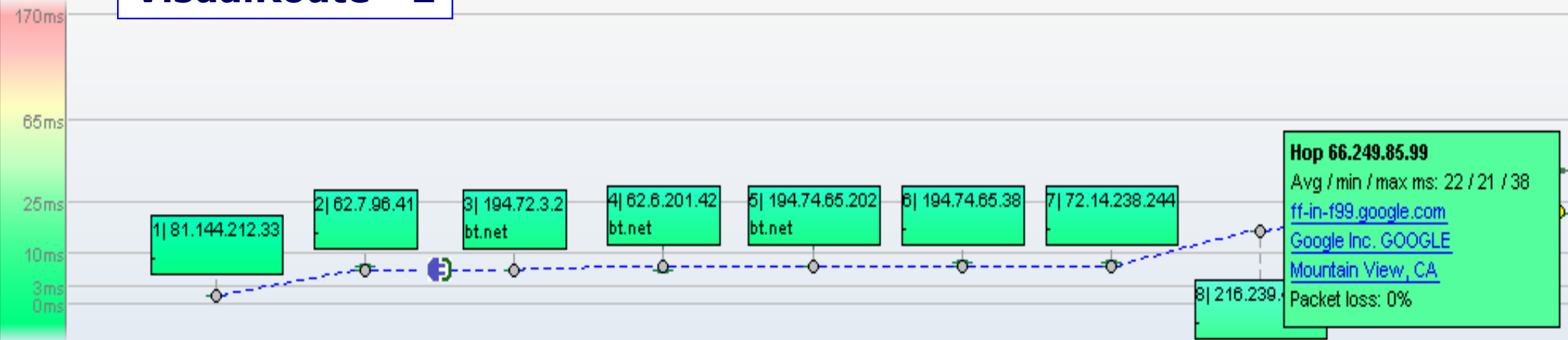
Route Graph



Route Table



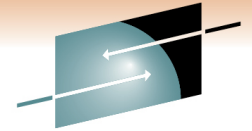
VisualRoute - 2



Hop	%Loss	IP Address	Node Name	Location	Tzone	ms	Graph	Network
0		192.168.1.238	wdsgdw.wds.local	*			0	33 (private use)
1		81.144.212.33	-	(United Kingdom)	*	1		FTIP002842486 William Data Sy
2		62.7.96.41	-	(United Kingdom)	*	6		BTnet
3		194.72.3.2	core2-gig2-1.kingston.ukcc	Kingston, London, UK	*	6		PoP
4		62.6.201.42	core2-pos7-3.ealing.ukcor	Ealing, UK	*	6		Infrastructure
5		194.74.65.202	core2-pos10-0.redbus.ukc	(United Kingdom)	*	7		Private Circuit Customer Networ
6		194.74.65.38	-	(United Kingdom)	*	7		Private Circuit Customer Networ
7		72.14.238.244	-	Mountain View, CA		7		Google Inc. GOOGLE
8		216.239.43.91	-	Mountain View, CA		16		Google Inc. GOOGLE
9		72.14.232.209	-	Mountain View, CA		22		Google Inc. GOOGLE
10		66.249.85.99	www.google.co.uk	Mountain View, CA		21		Google Inc. GOOGLE

Roundtrip time to www.google.co.uk, average = 21ms, min = 21ms, max = 21ms -- 08-Jan-2007 10:38:43 (Collapse Table)

Learn more at:
<http://www.visualroute.com>



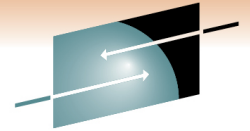
PingPlotter

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

0 - 200
201 - 500
501 and up

Hop	PL%	IP	DNSName	Avg	Cur	Graph
1		81.144.212.33	1	1	
2		62.7.96.41	6	6	
3		194.72.3.2	core2-gig2-1.kingston.ukcore.bt.net	6	6	
4		62.6.201.42	core2-pos7-3.ealing.ukcore.bt.net	7	7	
5		194.74.65.202	core2-pos10-0.redbus.ukcore.bt.net	7	7	
6		194.74.65.38	7	8	
7		72.14.238.244	7	7	
8		216.239.43.91	16	16	
9		72.14.232.209	21	21	
10		66.249.85.99	ff-in-f99.google.com	21	22	

Round Trip: 21 22



TRACEROUTE –Alternatives

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.

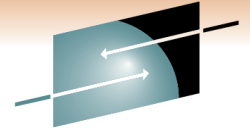
Basically, this is a “proxy” service ; the remote site issuing the test on your behalf.

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 DEV
TSO NETSTAT TCP TCPIP

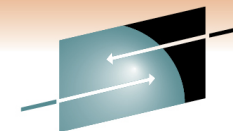
TSO NETSTAT SOCK
TSO NETSTAT ROUTE

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. . .



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
DestAddr: 0.0.0.0
Packet Trace Setting:
  Protocol: 253
  SrcPort: *
  IpAddr: *
Multicast Specific:
  Multicast Capability: Y
  Group                RefCr
  -----
  224.0.0.1            00000
  
```

Link Statistics:

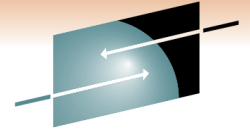
```

BytesIn
Inbound Packets
Inbound Packets In Error
Inbound Packets Discard
Inbound Packets with No
  
```

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
  
```



NETSTAT (Windows)

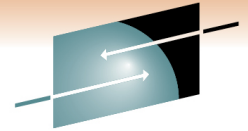
Usage: **netstat** [-a] [-b] [-e] [-n] [-o] [-p proto] [-r] [-s] [-v] [interval]

- 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

Active Connections

Proto	Local Address	Foreign Address	State
TCP	wdsgdw:epmap	0.0.0.0:0	LISTENING
TCP	wdsgdw:microsoft-ds	0.0.0.0:0	LISTENING
TCP	wdsgdw:1028	0.0.0.0:0	LISTENING
TCP	wdsgdw:1241	0.0.0.0:0	LISTENING
TCP	wdsgdw:10110	0.0.0.0:0	LISTENING
UDP	wdsgdw:microsoft-ds	*:*	
UDP	wdsgdw:isakmp	*:*	
UDP	wdsgdw:1033	*:*	
UDP	wdsgdw:4500	*:*	
UDP	wdsgdw:ntp	*:*	
UDP	wdsgdw:1900	*:*	



DNS . . .

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).

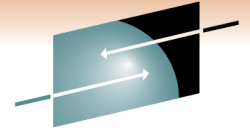
The IP address returned is then used to

**** HOSTS file from Windows :-
(C:\WINDOWS\system32\drivers\etc)**

127.0.0.1	localhost
192.168.1.45	lizzie
192.168.1.45	wds.local
192.168.1.45	wds
192.168.1.43	wdsnfs

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!



DNS . . .

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*.

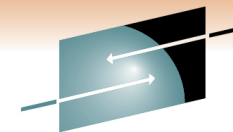
Take www.google.co.uk ...

- . 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**
or **command**

←(cf z/OS "Resolver")

set option

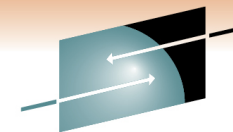
all	[no]debug	[no]d2	[no]defname
[no]recurse	[no]search	[no]vc	domain=NAME
srchlist=N1[/N2/.../N6]		root=NAME	retry=x
timeout=X	type=X	querytype=X	class=X
[no]msxfr	ixfrver=X		

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.



NSLOOKUP (Windows)

```
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
```

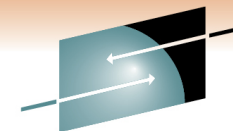
```
ttl = 60 (1 min)
```

```
-----  
Non-authoritative answer:
```

```
←----- ( Retrieved from a cache! )
```

```
Name: www.google.co.uk.uk.willdata.com
```

```
Address: 212.69.199.183
```



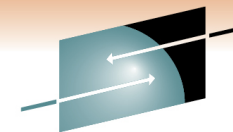
DIG

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**

>

dig @lizzie www.google.com any

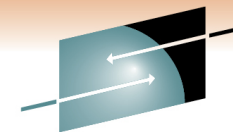
```
; <<>> DiG 9.3.1 <<>> @lizzie www.google.com any
; (1 server found) ; global options: printcmd ; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60773
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 3

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

;; ANSWER SECTION:
www.google.com.                 86400      IN          CNAME      www.l.google.com.

;; 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
```



WHOIS

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 Markmoni

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.

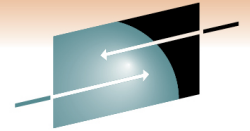
Name servers:

ns1.google.com

ns2.google.com

ns3.google.com

ns4.google.com



Pchar

Estimates bandwidth, latency and packet loss on network links.

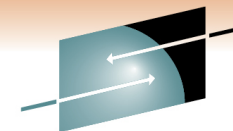
This is a re-working of the “pathchar” utility, written by Van Jacobson and, like traceroute, is based on repeated packet transmission and TTL variation (it can use ICMP or UDP).

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 - ./pchar www.google.co.uk

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
stddev rtt = 0.003212, stddev b = 0.000004

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
stddev rtt = 0.009218, stddev b = 0.000011

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
stddev rtt = 0.008008, stddev b = 0.000010

Partial queueing: avg = 0.001486 ms (667 bytes)

Hop char: rtt = 0.040220 ms, bw = --.--- kbps

Hop queueing: avg = -0.000850 ms (0 bytes)

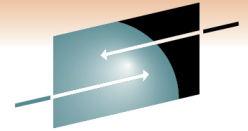
3: 194.72.3.66 (core2-gig10-1.kingston.ukcore.bt.net)

???

-

process hangs at this point!

This example shows a "pchar" test across a path where icmp responses are **not** allowed.



Pchar - ./pchar 192.168.1.8 (a local address)

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)

Path length: 1 hops

Path char: rtt = 10.792415 ms r2 = 0.157013

Path bottleneck: 2374.706954 Kbps

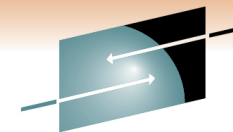
Path pipe: 3203 bytes

Path queueing: average = 0.015037 ms

Start time: Thu Feb 1 09:07

End time: Thu Feb 1 09:14

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 bottleneck	= "bottleneck" (achieved) bandwidth
Path pipe	= Bandwidth-Delay Product = traffic "on the wire" (cf RWIN buffer)



Pchar

Remember:

ICMP may be restricted over the test path

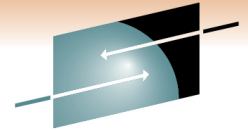
Not all platforms have the same controls or defaults

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/>



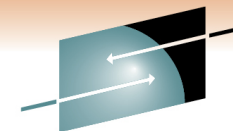
Netcat

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

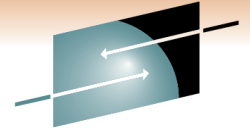
```
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound:  nc -l -p port [options] [hostname] [port]
options:
    -d                detach from console, background mode
    -e prog           inbound program to exec [dangerous!!]
    -g gateway        source-routing hop 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
    -z                zero-I/O mode

port numbers can be individual or range
```

Learn more at:

<http://netcat.sourceforge.net/>

<http://nmap.org/ncat/>

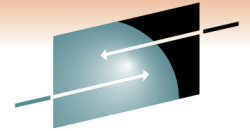


Netcat - Retrieve page from web server

```
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=DBxO29wrWXh5ex5E;
  expires=Sun, 17-Jan-2038 19:14:07 GMT;
  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
```

192.168.27.10

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

192.168.27.10

Active Connections

```
Proto Local Address Foreign Address
TCP wds-gdw:ftp
TCP wds-gdw:telnet
TCP wds-gdw:ep
TCP wds-gdw:msn
TCP wds-gdw:10
TCP wds-gdw:53
TCP wds-gdw:10
. . . .
```

```
C:\>nc 192.168.27.10 23
```

192.168.27.50

```
Microsoft windows XP [Version 5.1.2600] . . .
```

```
C:\>ipconfig
```

```
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
```

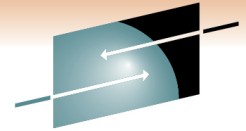
```
C:\>^C
```

```
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
```



iReasoning

iReasoning MIB Browser

File Edit Tools Help

Address: 192.168.1.231:161 Advanced... OID: .1.3.6.1.2.1.2 Go

SNMP MIBs

MIB Tree

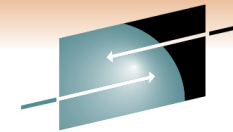
- RFC1213-MIB.iso.org.dod.internet.mgmt.mib-2
 - system
 - sysDescr
 - sysObjectID
 - sysUpTime
 - sysContact
 - sysName
 - sysLocation
 - sysServices
 - interfaces**
 - ifNumber
 - ifTable
 - ifEntry
 - at
 - ip
 - icmp
 - tcp
 - udp
 - egp
 - transmission
 - snmp

Name/OID	Value
.1.3.6.1.2.1.1.9.1.4.7	3
.1.3.6.1.2.1.1.9.1.4.8	3
.1.3.6.1.2.1.1.9.1.4.9	3
ifNumber.0	3
ifIndex.1	1
ifIndex.2	2
ifIndex.3	3
ifDescr.1	lo
ifDescr.2	eth0
ifDescr.3	sit0
ifType.1	softwareLoopback
ifType.2	ethernet-csmacd
ifType.3	131
ifMtu.1	16436
ifMtu.2	1500
ifMtu.3	1480
ifSpeed.1	10000000
ifSpeed.2	100000000
ifSpeed.3	0
ifPhysAddress.1	
ifPhysAddress.2	0x00 0x06 0x5B 0x37 0xF3 0x46
ifPhysAddress.3	
ifAdminStatus.1	up
ifAdminStatus.2	up
ifAdminStatus.3	down
ifOperStatus.1	up
ifOperStatus.2	up
ifOperStatus.3	down
ifInOctets.1	517417240
ifInOctets.2	3765775664
ifInOctets.3	0
ifInUcastPkts.1	1333084
ifInUcastPkts.2	
ifInUcastPkts.3	
ifInDiscards.1	
ifInDiscards.2	

Node Name	interfaces
OID	.1.3.6.1.2.1.2
Syntax	
Access	
Status	
DefVal	
Indexes	
Descr	

.iso.org.dod.internet.mgmt.mib-2.interfaces

Learn more at:
<http://www.ireasoning.com/>



IMPLEX

SNMP MIB Browser

ADCDPL

P390

TCPIP

14:48:16

Host Name 192.168.1.231

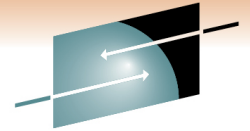
Community public

MaxRequest 128

Object	Value
system	
interfaces	
ifNumber	3
ifTable	
ifEntry	
ifIndex	
.1	1
.2	2
.3	3
ifDescr	(1)
ifType	(1)
ifMtu	(1)
ifSpeed	(1)
ifPhysAddress	(1)
ifAdminStatus	(1)
ifOperStatus	(1)
ifLastChange	(1)
ifInOctets	(1)
ifInUcastPkts	(1)
ifInNUcastPkts	(1)
ifInDiscards	(1)
ifInErrors	(1)
ifInUnknownProtos	(1)
ifOutOctets	(1)
ifOutUcastPkts	(1)
ifOutNUcastPkts	(1)
ifOutDiscards	(1)
ifOutErrors	(1)
ifOutQLen	(1)
ifSpecific	(1)
at	
ip	
icmp	
tcp	

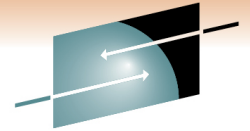
Objects 265

7671



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.
- **ETHERREAL** is a packet analyzer based on TCPDUMP.
- **WIRESHARK** is the latest incarnation of ETHERAL
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.



"Wireshark"

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.27.10	192.168.1.18	SNMP	get-request
2	0.070429	192.168.1.18	192.168.27.10	SNMP	get-response
3	15.103364	192.168.27.10	192.168.27.1	DNS	Standard query PTR 1.27.168.192.in-addr.arpa
4	15.103965	192.168.27.1	192.168.27.10	DNS	Standard query response PTR my.router
5	15.105789	192.168.27.10	192.168.27.1	DNS	Standard query A www.btopenworld.com.uk.willdata.com
6	15.204957	192.168.27.1	192.168.27.10	DNS	Standard query response A 212.69.199.183
7	40.612179	192.168.27.10	192.168.27.1	DNS	Standard query PTR 1.27.168.192.in-addr.arpa
8	40.612799	192.168.27.1	192.168.27.10	DNS	Standard query response PTR my.router
9	43.353181	Draytek_d0:1a:a8	Broadcast	ARP	who has 192.168.27.50? Tell 192.168.27.1
10	52.344425	192.168.27.10	192.168.27.255	BROWSE	Local Master Announcement WDS-GDW, workstation, Server, NT workstation, Potential Browser, M
11	57.879698	192.168.27.10	192.168.27.1	DNS	Standard query A www.btopenworld.com.uk.willdata.com
12	57.880039	192.168.27.1	192.168.27.10	DNS	Standard query response A 212.69.199.183

Frame 3 (85 bytes on wire, 85 bytes captured)

Ethernet II, Src: Dell_d2:4d:66 (00:12:3f:d2:4d:66), Dst: Draytek_d0:1a:a8 (00:50:7f:d0:1a:a8)

Internet Protocol, src: 192.168.27.10 (192.168.27.10), dst: 192.168.27.1 (192.168.27.1)

Version: 4
Header length: 20 bytes

- Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
 - 0000 00.. = Differentiated Services Codepoint: Default (0x00)
 -0. = ECN-Capable Transport (ECT): 0
 -0 = ECN-CE: 0
- Total Length: 71
- Identification: 0x6b5d (27485)
- Flags: 0x00
 - 0... = Reserved bit: Not set
 - .0.. = Don't fragment: Not set
 - ..0. = More fragments: Not set
- Fragment offset: 0
- Time to live: 128
- Protocol: UDP (0x11)

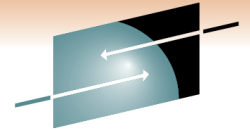
0000 00 50 7f d0 1a a8 00 12 3f d2 4d 66 08 00 45 00 .P.....?.Mf..E.
 0010 00 47 6b 5d 00 00 80 11 17 ed c0 a8 1b 0a c0 a8 .GK]....
 0020 1b 01 06 62 00 35 00 33 ed a6 00 01 01 00 00 01 ..b.5.3
 0030 00 00 00 00 00 00 01 31 02 32 37 03 31 36 38 031.27.168.
 0040 31 39 32 07 69 6e 2d 61 64 64 72 04 61 72 70 61 192.in-a ddr.arpa
 0050 00 00 0c 00 01

Internet Protocol (ip), 20 bytes | P: 12 D: 12 M: 0 Drops: 0

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 . . .



"Wireshark"

Filter: Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.27.10	192.168.1.18	SNMP	get-request
2	0.070429	192.168.1.18	192.168.27.10	SNMP	get-response
3	15.103364	192.168.27.10	192.168.27.1	DNS	Standard query PTR 1.27.168.192.in-addr.arpa
4	15.103965	192.168.27.1	192.168.27.10	DNS	Standard query response PTR my.router
5	15.105789	192.168.27.10	192.168.27.1	DNS	Standard query A www.btopenworld.com.uk.willdata.com
6	15.204957	192.168.27.1	192.168.27.10	DNS	Standard query response A 212.69.199.183
7	40.612179	192.168.27.10	192.168.27.1	DNS	Standard query PTR 1.27.168.192.in-addr.arpa
8	40.612799	192.168.27.1	192.168.27.10	DNS	Standard query response PTR my.router
9	43.353181	Draytek_d0:1a:a8	Broadcast	ARP	who has 192.168.27.50? Tell 192.168.27.1
10	52.344425	192.168.27.10	192.168.27.255	BROWSE	Local Master Announcement WDS-GDW, workstation, Server, NT workstation, Potential Browser, M.
11	57.879698	192.168.27.10	192.168.27.1	DNS	Standard query A www.btopenworld.com.uk.willdata.com
12	57.880039	192.168.27.1	192.168.27.10	DNS	Standard query response A 212.69.199.183

⊕ Frame 3 (85 bytes on wire (85 bytes captured))

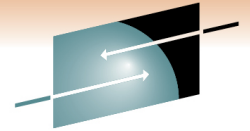
- ⊕ Ethernet II, Src: dell_d2:4d:66 (00:12:3f:d2:4d:66), Dst: Draytek_d0:1a:a8 (00:50:7f:d0:1a:a8)
- ⊕ Internet Protocol, Src: 192.168.27.10 (192.168.27.10), Dst: 192.168.27.1 (192.168.27.1)
- ⊖ User Datagram Protocol, Src Port: 1634 (1634), Dst Port: domain (53)
 - Source port: 1634 (1634)
 - Destination port: domain (53)
 - Length: 51
 - Checksum: 0xeda6 [correct]
- ⊖ Domain Name System (query)
 - [\[Response In: 4\]](#)
 - Transaction ID: 0x0001
 - ⊕ Flags: 0x0100 (standard query)
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - ⊖ Queries
 - ⊖ 1.27.168.192.in-addr.arpa: type PTR, class IN

```

0000  00 50 7f d0 1a a8 00 12 3f d2 4d 66 08 00 45 00  .P.....?.Mf..E.
0010  00 47 6b 5d 00 00 80 11 17 ed c0 a8 1b 0a c0 a8  .Gk]....
0020  1b 01 06 62 00 35 00 33 ed a6 00 01 01 00 00 01  ..b.5.3...
0030  00 00 00 00 00 00 01 31 02 32 37 03 31 36 38 03  .....1.27.168.
0040  31 39 32 07 69 6e 2d 61 64 64 72 04 61 72 70 61  192.in-a ddr.arpa
0050  00 00 0c 00 01  .....
    
```

User Datagram Protocol (udp), 8 bytes | P: 12 D: 12 M: 0 Drops: 0

This image shows the UDP header . . .



"Wireshark"

The screenshot shows the Wireshark interface with a list of 12 captured packets. Packet 10 is highlighted in yellow, showing a BROWSE Local Master Announcement. Packet 11 is also highlighted, showing a DNS Standard query A for www.btopenworld.com.uk.willdata.com. Packet 12 is a DNS Standard query response A for 212.69.199.183.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.27.10	192.168.1.18	SNMP	get-request
2	0.070429	192.168.1.18	192.168.27.10	SNMP	get-response
3	15.103364	192.168.27.10	192.168.27.1	DNS	Standard query PTR 1.27.168.192.in-addr.arpa
4	15.103965	192.168.27.1	192.168.27.10	DNS	Standard query response PTR my.router
5	15.105789	192.168.27.10	192.168.27.1	DNS	Standard query A www.btopenworld.com.uk.willdata.com
6	15.204957	192.168.27.1	192.168.27.10	DNS	Standard query response A 212.69.199.183
7	40.612179	192.168.27.10	192.168.27.1	DNS	Standard query PTR 1.27.168.192.in-addr.arpa
8	40.612799	192.168.27.1	192.168.27.10	DNS	Standard query response PTR my.router
9	43.353181	Draytek_d0:1a:a8	Broadcast	ARP	who has 192.168.27.50? Tell 192.168.27.1
10	52.344425	192.168.27.10	192.168.27.255	BROWSE	Local Master Announcement WDS-GDW, workstation, Server, NT workstation, Potential Browser, M.
11	57.879698	192.168.27.10	192.168.27.1	DNS	Standard query A www.btopenworld.com.uk.willdata.com
12	57.880039	192.168.27.1	192.168.27.10	DNS	Standard query response A 212.69.199.183

The detailed view of packet 11 shows the following information:

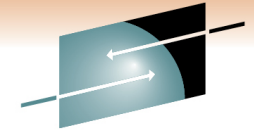
- Ethernet II, Src: Dell_d2:4d:66 (00:12:3f:d2:4d:66), Dst: Draytek_d0:1a:a8 (00:50:7f:d0:1a:a8)
- Internet Protocol, Src: 192.168.27.10 (192.168.27.10), Dst: 192.168.27.1 (192.168.27.1)
- User Datagram Protocol, Src Port: 1634 (1634), Dst Port: domain (53)
- Domain Name System (query)
 - Response In: 4
 - Transaction ID: 0x0001
 - Flags: 0x0100 (Standard query)
 - 0... .. = Response: Message is a query
 - .000 0... .. = opcode: standard query (0)
 -0. = Truncated: Message is not truncated
 -1 = Recursion desired: Do query recursively
 -0... .. = Z: reserved (0)
 -0 = Non-authenticated data OK: Non-authenticated data is unacceptable
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - Queries
 - 1.27.168.192.in-addr.arpa: type PTR, class IN

The hex dump at the bottom shows the raw data of the DNS query:

```

0010 00 47 80 3d 00 00 80 11 17 ed c0 a8 10 0a c0 a8 .GKJ....
0020 1b 01 06 62 00 35 00 33 ed a6 00 01 01 00 00 01 ...b.5.3
0030 00 00 00 00 00 00 01 31 02 32 37 03 31 36 38 03 .....1.27.168.
0040 31 39 32 07 69 6e 2d 61 64 64 72 04 61 72 70 61 192.in-a ddr.arpa
0050 00 00 0c 00 01 .....
    
```

This image shows the DATA; in this case a DNS Query.
(<http://www.wireshark.org/>)



"EXIGENCE"

Trace 132 : trace 1 TCPIP [Zos15]

Record	Time (GMT+00)	IP Address 1	Port 1	Direction	IP Address 2	Port 2
1	13:15:31.124	10.20.1.81	ftp	← TCP connect →	10.20.1.71	
2	13:15:31.136	10.20.1.81	ftp	→ TCP connect ←	10.20.1.71	
4	13:15:31.625	10.20.1.81	ftp	→ ftp-Ready ←	10.20.1.71	
6	13:15:31.854	10.20.1.81	ftp	→ ftp-Ready ←	10.20.1.71	
7	13:15:31.933	10.20.1.81	ftp	← ftp-User →	10.20.1.71	
9	13:15:33.335	10.20.1.81	ftp	→ ftp-Pwd needed ←	10.20.1.71	
10	13:15:33.347	10.20.1.81	ftp	← ftp-Password →	10.20.1.71	
12	13:15:35.151	10.20.1.81	ftp	→ ftp-Logged in ←	10.20.1.71	
13	13:15:35.184	10.20.1.81	ftp	← ftp-Type →	10.20.1.71	
14	13:15:35.265	10.20.1.81	ftp	→ ftp-OK ←	10.20.1.71	
15	13:15:35.558	10.20.1.81	ftp	← ftp-Port →	10.20.1.71	
16	13:15:35.597	10.20.1.81	ftp	→ ftp-OK ←	10.20.1.71	
17	13:15:35.613	10.20.1.81	ftp	← ftp-Retrieve →	10.20.1.71	
18	13:15:35.709	10.20.1.81	ftp-data	→ TCP connect ←	10.20.1.71	
19	13:15:35.712	10.20.1.81	ftp-data	← TCP connect →	10.20.1.71	
22	13:15:35.924	10.20.1.81	ftp	→ ftp-xfer starting ←	10.20.1.71	
23	13:15:36.010	10.20.1.81				
24	13:15:36.010	10.20.1.81				
26	13:15:36.075	10.20.1.81				
29	13:15:36.362	10.20.1.81				
31	13:15:36.383	10.20.1.81				
32	13:15:36.431	10.20.1.81				
33	13:15:36.470	10.20.1.81				
34	13:15:36.491	10.20.1.81				
36	13:15:36.514	10.20.1.81				

Exigence Trace 132 Record 22 [Zos15]

File Expanded Navigate View Window Help

HEX FMT TRN EBC

IP Header 4500006C DDC50000 40068607 0A140151 0A140147

TCP Header 0015043A 4E039655 52FBB40A 80187FE4 995B0000 0101080A BCA...

FTP data Already open; transfer starting

```

+0000 31323520 53656E64 696E6720 64617461 *125 Sending data*
+0010 20736574 20574453 2E465450 54455354 * set WDS.FTPTEST*
+0020 2E53495A 4531364B 20464958 72656366 *.SIZE16K FIXref*
+0030 6D203130 32340D0A *m 1024.. *
```

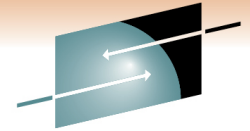

Exigence Trace 132 Record 22 [Zos15]

Window Help View

```

Transmission Control Protocol (TCP) header
0000 0015 (21) Source Port
0002 043A (1082) Destination Port
0004 4E039655 (1308857941) Sequence Number
0008 52FBB40A (1392227338) Acknowledgement Number
000C 80 (8) Length of TCP header in words (bits 0-3)
      ---- xxxx Reserved for future use (bits 4-7)
000D 18 Code bits
      xx-- ---- Reserved for future use
      ---1 ---- (ACK) Acknowledgement field is relevant
      ---- 1---- (PSH) Don't wait for buffer full to send data
000E 7FE4 (32740) Most bytes
      ---- ----
```

This image shows the equivalent displays in EXIGENCE; in this case for an FTP session.
(<http://www.willdata.com/>)



"ZEN Trace and Solve"

New	TraceID	Status	Taken	Description	Userid	Entries	
	0001	Taken	16 Feb 2011 11:47	Test Trace	TONYA	220	X
	0002	Taken	16 Feb 2011 11:49	EE Trace	TONYA	178	X
	0003	Taken	17 Feb 2011 12:18	Interface LNKOSA48	TONYA	18	X
	0004	Taken	17 Feb 2011 16:04	Interface LNKOSA48	TONYA	24	X

Trace 0001 Started 16-02-2011 at 11:47: Packets traced 220

Packet Time	Packet Length	Protocol	IP Address	IP Address	Ports	Window Seq No.	ACK Seq No.	Window size
000001	11:47:05.23153	64	TCP	10.5.1.11	10.5.1.11	6121 2499	B0AB74F9	397F08AF 32731
000002	11:47:05.24325	64	TCP	10.5.1.11	10.5.1.11	6121 2499	397F08AF	B0AB7505 32756
000003	11:47:05.24968	706	TCP	10.5.1.11	10.5.1.11	6121 2499	B0AB7505	397F08BB 32756
000004	11:47:05.25422	78	UDP	10.20.1.133	255.255.255.255	137 137		
000005	11:47:05.57010	217	TCP	192.168.5.103	10.5.1.11	50113 27000	10B9AD90	EFA95BAA 32660
000006	11:47:05.57149	576	TCP	192.168.5.103	10.5.1.11	50113 27000	10B9AE35	EFA95BAA 32660
000007	11:47:05.57149	576	TCP	192.168.5.103				
000008	11:47:05.57149	576	TCP	192.168.5.103				
000009	11:47:05.57149	292	TCP	192.168.5.103				
000010	11:47:05.57338	52	TCP ACK	10.5.1.10				
000011	11:47:05.57462	52	TCP ACK	10.5.1.11				
000012	11:47:05.65033	52	TCP ACK	192.168.5.103				
000013	11:47:05.65191	52	TCP ACK	192.168.5.103				
000014	11:47:05.65203	52	TCP ACK	192.168.5.103				
000015	11:47:06.48634	205	UDP	192.168.23.5				
000016	11:47:06.48647	241	UDP	10.5.1.241				
000017	11:47:06.48663	241	UDP	10.5.1.241				
000018	11:47:06.82501	56	TCP	10.5.1.11				
000019	11:47:06.83992	576	TCP	10.5.1.11				
000020	11:47:06.84038	52	TCP ACK	10.5.1.11				
000021	11:47:06.84128	106	TCP	10.5.1.11				
000022	11:47:06.84200	52	TCP ACK	10.5.1.11				
000023	11:47:06.84388	56	TCP	10.5.1.10				
000024	11:47:06.84489	576	TCP	10.5.1.10				
000025	11:47:06.84489	106	TCP	10.5.1.10				
000026	11:47:06.84594	56	TCP	10.5.1.11				
000027	11:47:06.84662	630	TCP	10.5.1.11				
000028	11:47:06.84689	52	TCP ACK	10.5.1.11				
000029	11:47:06.85004	52	TCP ACK	10.5.1.10				
000030	11:47:06.85288	52	TCP ACK	10.5.1.10				

Trace 0001 Entry 000005 Header Expansion

+09	Protocol	TCP
+0A	Checksum	8216
+0C	Source IP Address	10.5.1.11
+10	Target IP Address	192.168.5.103
TCP Header		
+00	Source Port	27000
+02	Destination Port	50113
+04	Sequence Number	10B9AD90
+08	Ack Sequence Number	EFA95BAA
+0C	Header Length	32 (8 words)
+0D	Flags	PSH ACK
+0E	Window Size	32660
+10	TCP Checksum	35069
+12	Urgent Pointer	0
Packet Data Data Length 165)		

EBCDIC Text Top Previous Next Bottom

```

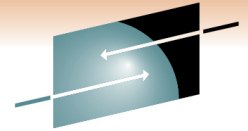
+0000 48545450 2F312E31 20323030 204F4B0D *HTTP/1.1 200 OK.*
+0010 0A446174 653A2057 65642C20 31362046 *.Date: Wed, 16 Feb
+0020 65622032 30313120 31313A34 373A3035 *eb 2011 11:47:05*
+0030 20474D54 0D0A5365 72766572 3A205A65 * GMT..Server: Ze*
+0040 6E0D0A43 61636865 2D436F6E 74726F6C *. Cache-Control*
+0050 3A206E6F 2D636163 68650D0A 436F6E6E *: no-cache..Conn*
+0060 65637469 6F6E3A20 4B656570 2D416C69 *action: Keep-Alive*
+0070 76650D0A 436F6E74 656E742D 54797065 *ve..Content-Type*
+0080 3A207465 78742F68 746D6C0D 0A436F6E *: text/html..Con*
+0090 74656E74 2D4C656E 6774683A 20313831 *tent-Length: 181*
+00a0 32 *2*
```

16:27:07

Show Elapsed Expand No Acks No Frags Goto Entry 1 Pair Select

16:27:21

16:31:02



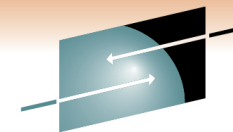
"ZEN Trace and Solve"

The screenshot displays the ZEN Trace and Solve application interface. At the top, a table lists several traces:

TraceID	Status	Taken	Description	Userid	Entries
0001	Taken	16 Feb 2011 11:47	Test Trace	TONYA	220
0002	Taken	16 Feb 2011 11:49	EE Trace	TONYA	178
0003	Taken	17 Feb 2011 12:18	Interface LNKOSA48	TONYA	18
0004	Taken	17 Feb 2011 16:04	Interface LNKOSA48	TONYA	24

The main window shows a detailed view of Trace 0001, titled "Trace 0001 Started 16-02-2011 at 11:47: Packets traced 220". It displays a list of packets with their times and lengths, along with expanded header information for each packet. The interface includes navigation buttons at the bottom such as "Show Elapsed", "Contract", "No Acks", "No Frags", "Goto Entry", "Pair Select", "Top", "Previous", "Next", and "Bottom".

ZTS - Exigence in the ZEN Framework.
(<http://www.willdata.com/>)

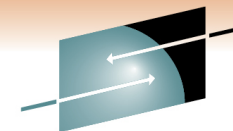


Network & Security testers

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

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

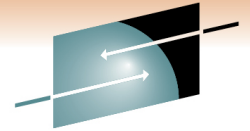
Use responsibly – Use with care !



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 Pcba 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)
```

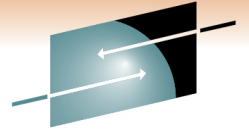
(NB. This sample has been edited to fit !)



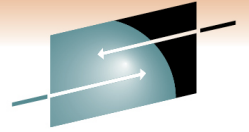
Outline Steps:

- Check the stack – “**ping**” local loopback
- “**ping**” the remote host/server name
- “**ping**” with IPaddress – the DNS may be down
- If “ping” fails “**tracert**” - 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 !