



SHARKFEST '14

WIRESHARK DEVELOPER AND USER CONFERENCE

JUNE 16-20 2014 · DOMINICAN UNIVERSITY

Definitive Diagnostic Data

<http://www.skendric.com/seminar/>

Stuart Kendrick
Sustaining Engineer
EMC Isilon

Definitive Diagnostic Data

A Rapid Problem Resolution® perspective

Advance7 is a consulting outfit which helps customers resolve critical Problems – they put an analyst at your site to coordinate your staff plus vendors to fix the issue, using the RPR methodology.

I don't work for Advance7, and I have only a rudimentary grasp of RPR – I've read Paul Offord's book and attended Advance7's two-day *Foundations in RPR* seminar.

On the other hand, I have employed bits & pieces of RPR on the job, and I've found it effective – in fact, any time I get near a trouble-shooting job, I try to employ as much of RPR as I can manage.

A signature feature of RPR is its concept of *Definitive Diagnostic Data* (D³).

Over the next hour+, I plan to sketch my understanding of D³, focused particularly on the concrete technique of *markers*, which thread their way through D³.

Mechanics

Talk

- I encourage interactivity
- If you want to contribute, feel free to interrupt me
- Or raise your hand, and I'll call on you
- I'm good with either approach

This deck visible at <http://www.skendric.com/seminar/>

Me

Multi-disciplinary IT trouble-shooter / Root Cause Analysis

<http://www.skendric.com>

sbk@cornella	<i>student</i>	1981
stuart@cpvax5 (Science Applications Inc)	<i>programmer</i>	1984
sbk@cornellc.cit.cornell.edu	<i>desktop / server</i>	1985
stuart.kendrick@med.cornell.edu	<i>server / network</i>	1991
skendric@fhcrc.org	<i>multidisciplinary</i>	1993
stuart.kendrick@isi lon dot com	<i>sustaining engineer</i>	2013

IT Architect | ITIL Problem Manager | Problem Analyst | Device Monitoring | Transport

Geeky Highlights

PL/1 on IBM mainframes	<i>Cornell University</i>	<i>Ithaca</i>	1981
FORTRAN on CRAY-1	<i>SAIC</i>	<i>San Diego</i>	1984
Terak, DisplayWriter, IBM PC, Macintosh	<i>Cornell University</i>	<i>Ithaca</i>	1985
Netware, Corvus Omninet, TCP-IP / IPX / AppleTalk	<i>Cornell University</i>	<i>Ithaca</i>	1988
AppleShare, QuickMail, Farallon, NRC, Cisco, Sniffers	<i>Cornell Medical College</i>	<i>Manhattan</i>	1991
Solaris, Windows, Linux, Perl, SNMP, Wireshark, Cisco, Fluke	<i>FHCRC</i>	<i>Seattle</i>	1993
OneFS	<i>EMC Isilon</i>	<i>Seattle</i>	2013

Geek credentials: I missed punch-cards by one semester ... grew up on shared machines (IBM and Cray) ... my first network ran at 1Mb/s over Cat 2 (Corvus Omninet) carrying IPX + AppleTalk with IP encapsulated in both. I bored a vampire tap (once) ... my first analyzer was a Network General Toshiba 286 laptop ... and alpha versions of EtherPeek

Recruiting

I attend SharkFest for a lot of reasons ...

But one of them is recruiting.

Isilon

If you would like to hear what it is like to work at Isilon, I would enjoy sharing the pros, and the cons, of working in this space.

You may not be interested in changing jobs right now – from my point of view, I would still enjoy talking with you – perhaps your situation will change in a year or two. *Isilon invests long-term in staff; a multi-year courtship suits our style just fine.*

Richly complex product, engineering-oriented company, plenty of difficult problems to solve. Global company, numerous locations, and once you're sufficiently senior, plenty of flexibility in terms of operating remotely, telecommuting, and visiting a base office every quarter or so.

FHCRC

My old position at the Hutch is still open ... Problem Manager / Problem Analyst / emcee of RCAs, with oversight over Change Management and post-mortems arising from Incidents. Also, Network Manager (four techs, physical layer, Ethernet/IP/WiFi transport, firewall operations, Internet connectivity, voice.)

Come find me during a break or in the evening. Professional networking is a good thing.

So What Is *Definitive Diagnostic Data*?

At a first pass, D³ means inserting *markers* into the data stream you are capturing

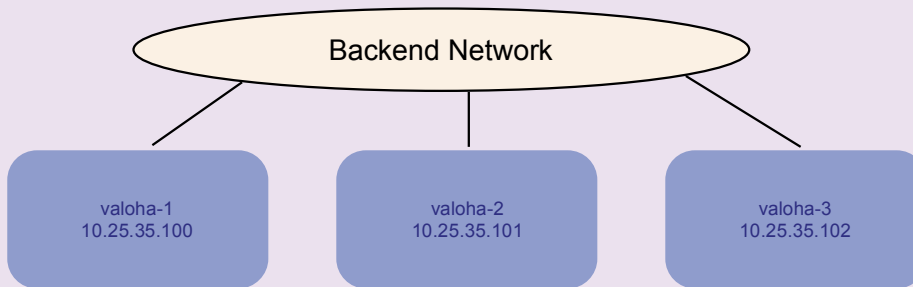
- Markers help you find the section of trace where some interesting event occurred
- Markers function as in-band documentation on what happened and when
- Markers contribute to concreteness

I know event xyz occurred after this point here and not before

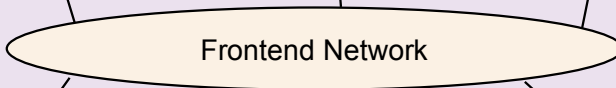
Thus I know I can ignore this chunk of the trace and focus my attention on this other chunk

```
/etc/resolv.conf
search widgets.company.com
nameserver 127.0.0.1
nameserver 10.25.34.74
nameserver 10.25.34.24

/etc/ntp.conf
server 10.35.34.24
server 10.25.34.94
```

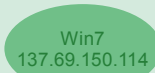


v7.1.0.0 + Patch 122486



Windows 2003 x64

```
resolv.conf net time/queriesntp
10.25.34.24 10.25.34.24,10.25.34.94
10.25.34.74
10.25.34.94
10.25.34.134
```



Each DC runs Domain Controller and DNS services
No external forwarding

widgets.com

Automated drive mapping

```
echo off
REM This script maps drives to a cluster, unmaps then, sleeps, and then repeats
REM The idea is to trigger intermittent cluster accessibility problems and to
REM record the time of those events, to be correlated with data capture efforts
REM running in parallel outside this script
REM
REM V   Who   When   What
REM -----
REM 1.2.0 skendric 2014-05-30 Record affinitized DC correctly
REM 1.0.1 skendric 2014-05-14 Record affinitized DC to log file
REM 1.0.0 skendric 2014-04-29 First version
```

```
REM Generically useful startup stuff
setlocal
setlocal ENABLEDELAYEDEXPANSION
```

```
REM Assign local variables
set node1=10.25.35.100
set node2=10.25.35.101
set node3=10.25.35.102
set sleepLong=10
set sleepShort=5
set usage=usage: cycle-drive-mapping {user@domain} {password} [output file]
```

```
REM Locate binaries
set findCmd=C:\Windows\System32\find.exe
set klistCmd=C:\Windows\System32\klist.exe
set netCmd=c:\Windows\System32\net.exe
set nltestCmd=c:\Windows\System32\nltest.exe /sc_query:safra.com.br
set sleepCmd=c:\temp\sleep.exe
set wteeCmd=c:\temp\wtee.exe -a
set klistCmd=c:\windows\system32\klist.exe
```


Automated drive mapping

```
REM Grab command-line parameters
set user=%1
set password=%2
set output=%3

REM Sanity check
if not defined user (
    echo Must specify user
    echo %usage%
    exit /B 1
)
if not defined password (
    echo Must specify password
    echo %usage%
    exit /B 1
)

REM Assign defaults
if not defined output set output=c:\temp\cycle-drive-mapping.txt
```

Automated drive mapping

REM Loop forever

:BEGIN

echo. 2>&1 | %wteeCmd% %output%

echo. 2>&1 | %wteeCmd% %output%

echo. 2>&1 | %wteeCmd% %output%

echo ===== 2>&1 | %wteeCmd% %output%

eecho %date% %time%

echo Affinitized DC:

%nltestCmd% | %findCmd% "Trusted" 2>&1 | %wteeCmd% %output%

echo Purging Kerberos ticket 2>&1 | %wteeCmd% %output%

%klistCmd% purge

echo Mapping drives 2>&1 | %wteeCmd% %output%

echo Mapping x: to %node1% at %date% %time% 2>&1 | %wteeCmd% %output%

%netCmd% use x: \\%node1%\ifs /user:%user% %password% 2>&1 | %wteeCmd% %output%

echo Mapping y: to %node2% at %date% %time% 2>&1 | %wteeCmd% %output%

%netCmd% use y: \\%node2%\ifs /user:%user% %password% 2>&1 | %wteeCmd% %output%

echo Mapping z: to %node3% at %date% %time% 2>&1 | %wteeCmd% %output%

%netCmd% use z: \\%node3%\ifs /user:%user% %password% 2>&1 | %wteeCmd% %output%

echo Sleeping for %sleepShort% 2>&1 | %wteeCmd% %output%

%sleepCmd% %sleepShort% 2>&1 | %wteeCmd% %output%

echo Deleting mappings at %date% %time% 2>&1 | %wteeCmd% %output%

%netCmd% use /del x: 2>&1 | %wteeCmd% %output%

%netCmd% use /del y: 2>&1 | %wteeCmd% %output%

%netCmd% use /del z: 2>&1 | %wteeCmd% %output%

echo Sleeping for %sleepLong% 2>&1 | %wteeCmd% %output%

%sleepCmd% %sleepLong% 2>&1 | %wteeCmd% %output%

goto BEGIN

Skew clock on domain controller

The image shows a Wireshark capture of network traffic. The main pane displays a list of packets. Packet 447 is highlighted in black and is an ICMP message: "Destination unreachable (Port unreachable)". The packet details pane below shows the structure of this ICMP message, including the Ethernet II header, IP header, and UDP header. The data field of the UDP header is circled in red and contains the following text:

```
0000 00 50 56 a1 2f b3 00 50 56 a1 6b b1 08 00 45 00  IPv4..P.k...E.  
0010 00 3e 9d ae 00 00 40 11 83 53 0a 19 23 64 0a 19  .>....@. .S.#0  
0020 22 18 cf ce 02 9a 00 2a 59 e9 76 61 6c 6f 68 6  ".....* y.valoha  
0030 2d 31 3a 20 53 6b 65 77 69 6e 67 20 63 6c 6f 6  -1: Skewing cloc  
0040 6b 20 6f 6e 20 61 64 6d 74 30 31 0a                k on adm t01.
```

Drive mapping failing now

The image shows a Wireshark network traffic capture. The main pane displays a list of network packets. Packet 999 is highlighted in red and shows a TCP RST, ACK from 10.25.34.24 to 10.25.35.100. Packet 1003 is highlighted in blue and shows an ICMP message: Destination unreachable (Port unreachable). The packet details pane for packet 1002 is expanded, showing Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Data (26 bytes). The data field contains a hex dump and a corresponding ASCII representation. A red circle highlights the ASCII text: ".6....@. .-...". "6...." Y.Drive mapping failing now.

No.	Time	Bytes	Source	Destination	Protocol	Info
992	0.002216000	74	10.25.35.100	10.25.34.24	TCP	53952 > 88 [SYN] Seq=0 win=65535 Len=0 MSS=1460 WS=64 SACK
993	0.003200000	78	10.25.34.24	10.25.35.100	TCP	88 > 53952 [SYN, ACK] Seq=0 Ack=1 win=64240 Len=0 MSS=1460
994	0.000020000	66	10.25.35.100	10.25.34.24	TCP	53952 > 88 [ACK] Seq=1 Ack=1 win=131712 Len=0
995	0.000033000	311	10.25.35.100	10.25.34.24	KRB5	AS-REQ
996	0.000451000	166	10.25.34.24	10.25.35.100	KRB5	KRB Error: KRB5KRB_AP_ERR_SKEW
997	0.000048000	66	10.25.35.100	10.25.34.24	TCP	53952 > 88 [FIN, ACK] Seq=246 Ack=101 win=131712 Len=0
998	0.000082000	66	10.25.34.24	10.25.35.100	TCP	88 > 53952 [ACK] Seq=101 Ack=247 win=63995 Len=0
999	0.000044000	60	10.25.34.24	10.25.35.100	TCP	88 > 53952 [RST, ACK] Seq=101 Ack=247 win=0 Len=0
1000	0.001368000	97	10.25.35.100	10.25.34.52	SMB	Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
1001	0.199512000	60	10.25.34.52	10.25.35.100	TCP	3747 > 445 [ACK] Seq=1216 Ack=851 win=63390 Len=0
1002	1.212883000	68	10.25.35.100	10.25.34.24	UDP	Source port: 52790 destination port: 666
1003	0.000269000	96	10.25.34.24	10.25.35.100	ICMP	Destination unreachable (Port unreachable)
1004	1.099592000	60	10.25.34.52	10.25.35.100	TCP	3747 > 445 [FIN, ACK] Seq=1216 Ack=851 win=63390 Len=0
1005	0.000019000	54	10.25.35.100	10.25.34.52	TCP	445 > 3747 [ACK] Seq=851 Ack=1217 win=65535 Len=0
1006	0.000102000	54	10.25.35.100	10.25.34.52	TCP	445 > 3747 [FIN, ACK] Seq=851 Ack=1217 win=65535 Len=0
1007	0.000160000	60	10.25.34.52	10.25.35.100	TCP	3747 > 445 [ACK] Seq=1217 Ack=852 win=63390 Len=0
1008	3.141148000	122	10.25.35.100	10.25.34.74	LDAP	searchRequest(2) "<ROOT>" baseObject

Frame 1002: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0

- Ethernet II, Src: vmware_a1:6b:b1 (00:50:56:a1:6b:b1), Dst: vmware_a1:2f:b3 (00:50:56:a1:2f:b3)
- Internet Protocol Version 4, Src: 10.25.35.100 (10.25.35.100), Dst: 10.25.34.24 (10.25.34.24)
- User Datagram Protocol, Src Port: 52790 (52790), Dst Port: 666 (666)
- Data (26 bytes)

```
0000  00 50 56 a1 2f b3 00 50 56 a1 6b b1 08 00 45 00  .6....@. .-...k...E.
0010  00 36 8e dc 00 00 40 11 92 2d 0a 19 23 64 0a 19  .6....@. .-...k...E.
0020  22 18 ce 36 02 9a 00 22 59 e1 44 72 69 76 65 20  ".6...." Y.Drive
0030  6d 61 70 70 69 6e 67 20 66 61 69 6c 69 6e 67 20  mapping failing
0040  6e 6f 77 0a                                     now.
```

Fixing clock on domain controller

The image shows a Wireshark capture of network traffic. The main pane displays a list of packets. Packet 1563 is highlighted in red and shows a TCP RST, ACK from 88 to 53807. Packet 1573 is highlighted in green and shows an ICMP Echo (ping) failure from 10.25.34.24 to 10.25.35.100, with the message "destination unreachable (Port unreachable)".

No.	Time	Bytes	Source	Destination	Protocol	Info
1557	0.000083000	78	10.25.34.24	10.25.35.100	TCP	88 > 53807 [SYN, ACK] Seq=0 Ack=1 win=64240 Len=0 MSS=1460
1558	0.000012000	66	10.25.35.100	10.25.34.24	TCP	53807 > 88 [ACK] Seq=1 Ack=1 win=131712 Len=0
1559	0.000024000	1328	10.25.35.100	10.25.34.24	KRBS	TGS-REQ
1560	0.000222000	171	10.25.34.24	10.25.35.100	KRBS	KRB Error: KRB5KRB_AP_ERR_SKEW
1561	0.000029000	66	10.25.35.100	10.25.34.24	TCP	53807 > 88 [FIN, ACK] Seq=1263 Ack=106 win=131712 Len=0
1562	0.000131000	66	10.25.34.24	10.25.35.100	TCP	88 > 53807 [ACK] Seq=106 Ack=1264 win=62978 Len=0
1563	0.000034000	60	10.25.34.24	10.25.35.100	TCP	88 > 53807 [RST, ACK] Seq=106 Ack=1264 win=0 Len=0
1564	0.040354000	97	10.25.35.100	10.25.34.52	SMB	Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
1565	0.002857000	60	10.25.34.52	10.25.35.100	TCP	3842 > 445 [FIN, ACK] Seq=1216 Ack=851 win=63390 Len=0
1566	0.000018000	54	10.25.35.100	10.25.34.52	TCP	445 > 3842 [ACK] Seq=851 Ack=1217 win=65535 Len=0
1567	0.000066000	54	10.25.35.100	10.25.34.52	TCP	445 > 3842 [FIN, ACK] Seq=851 Ack=1217 win=65535 Len=0
1568	0.000112000	60	10.25.34.52	10.25.35.100	TCP	3842 > 445 [ACK] Seq=1217 Ack=852 win=63390 Len=0
1569	9.961483000	66	10.25.35.100	10.25.34.24	TCP	53803 > 445 [FIN, ACK] Seq=63 Ack=182 win=131712 Len=0
1570	0.000313000	66	10.25.34.24	10.25.35.100	TCP	445 > 53803 [FIN, ACK] Seq=182 Ack=64 win=64178 Len=0
1571	0.000018000	66	10.25.35.100	10.25.34.24	TCP	53803 > 445 [ACK] Seq=64 Ack=183 win=131712 Len=0
1572	9.574422000	69	10.25.35.100	10.25.34.24	UDP	source port: 56336 destination port: 666
1573	0.000176000	97	10.25.34.24	10.25.35.100	ICMP	destination unreachable (Port unreachable)

Frame 1572: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface 0
Ethernet II, Src: vmware_a1:6b:b1 (00:50:56:a1:6b:b1), Dst: vmware_a1:2f:b3 (00:50:56:a1:2f:b3)
Internet Protocol Version 4, Src: 10.25.35.100 (10.25.35.100), Dst: 10.25.34.24 (10.25.34.24)
User Datagram Protocol, Src Port: 56336 (56336), Dst Port: 666 (666)
Data (27 bytes)

```
0000  00 50 56 a1 2f b3 00 50 56 a1 6b b1 08 00 45 00  .PV./..P y k . .E.  
0010  00 37 91 d3 00 00 40 11 8f 35 0a 19 23 64 0a 19  .7....@. .5..#d.  
0020  22 18 dc 10 02 9a 00 23 59 e2 46 69 78 69 6e 6  ".....# Y.Fixing  
0030  20 74 69 6d 65 20 73 6b 65 77 20 6f 6e 20 61 64  time sk ew on ad  
0040  6d 74 30 31 0a                                     mt01.
```

How to insert markers into pcaps?

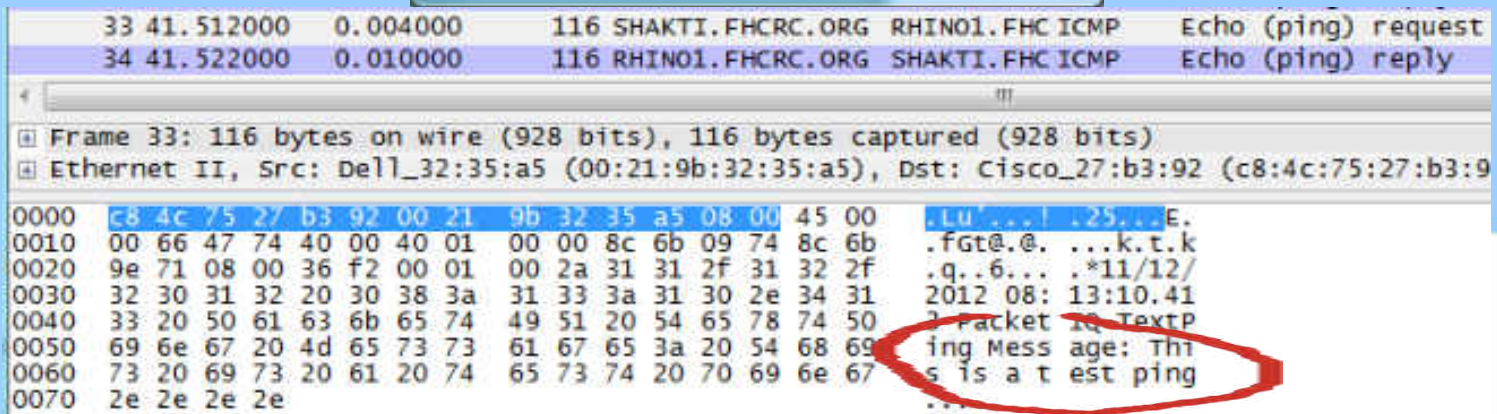
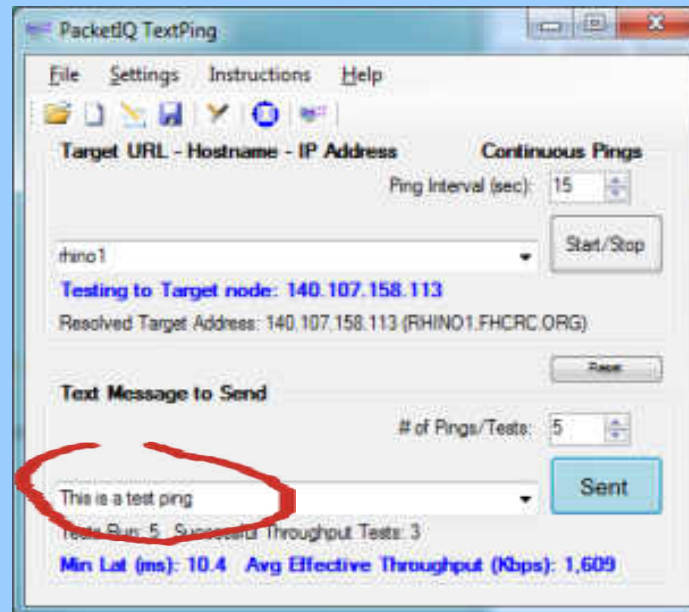
There are a lot of ways to do this

Building out this toolkit has taken me years

And colleagues continue to teach me new ways

TextPing

<http://www.packetiq.com/Tools/PacketIQ-TextPing.aspx>

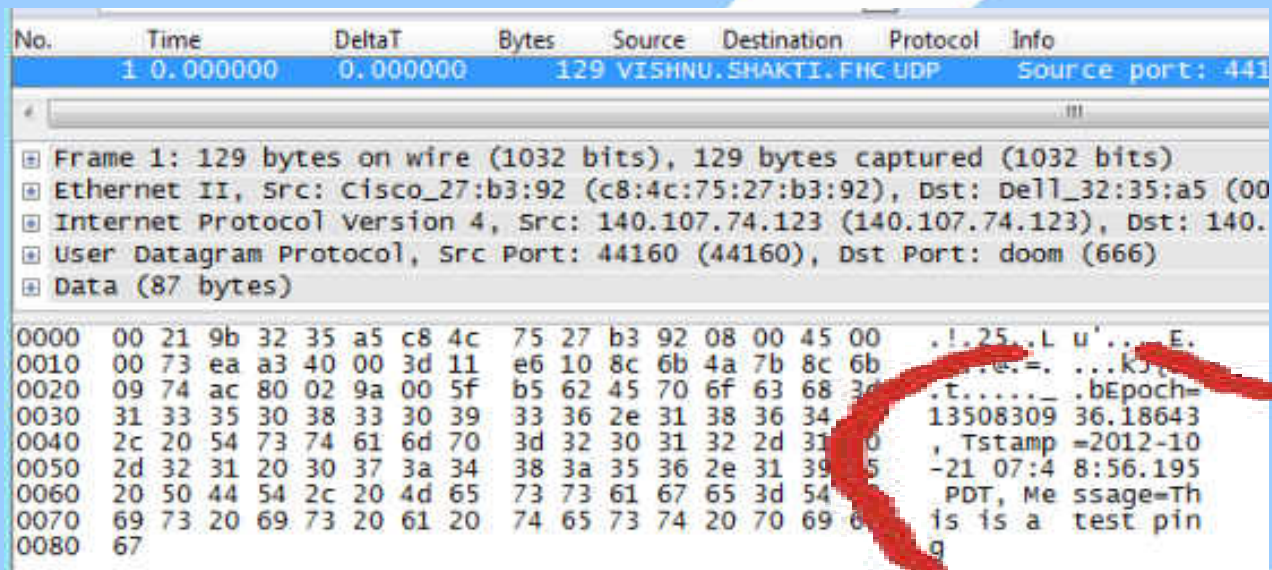


Send-UDP-Msg

<http://www.skendric.com/app>

Or write your own ... here's mine

```
vishnu> ./send-udp-msg -m "This is a test ping" rhino1 rhino2 rhino3  
vishnu>
```



No.	Time	DeltaT	Bytes	Source	Destination	Protocol	Info
1	0.000000	0.000000	129	VISHNU.SHAKTI.FHC	UDP		Source port: 44160
Frame 1: 129 bytes on wire (1032 bits), 129 bytes captured (1032 bits)							
Ethernet II, Src: Cisco_27:b3:92 (c8:4c:75:27:b3:92), Dst: Dell_32:35:a5 (00:0c:29:32:35:a5)							
Internet Protocol Version 4, Src: 140.107.74.123 (140.107.74.123), Dst: 140.107.74.123 (140.107.74.123)							
User Datagram Protocol, Src Port: 44160 (44160), Dst Port: doom (666)							
Data (87 bytes)							
0000	00 21 9b 32 35 a5 c8 4c		75 27 b3 92 08 00 45 00	.!.25..L u'...E.			
0010	00 73 ea a3 40 00 3d 11		e6 10 8c 6b 4a 7b 8c 6b	...e.=. ...k5			
0020	09 74 ac 80 02 9a 00 5f		b5 62 45 70 6f 63 68 3d	.t.....bEpoch=			
0030	31 33 35 30 38 33 30 39		33 36 2e 31 38 36 34	13508309 36.18643			
0040	2c 20 54 73 74 61 6d 70		3d 32 30 31 32 2d 31 00	, Tstamp =2012-10			
0050	2d 32 31 20 30 37 3a 34		38 3a 35 36 2e 31 39 05	-21 07:4 8:56.195			
0060	20 50 44 54 2c 20 4d 65		73 73 61 67 65 3d 54	PDT, Message=Th			
0070	69 73 20 69 73 20 61 20		74 65 73 74 20 70 69 00	is is a test pin			
0080	67			g			

So many techniques ...

Send a TCP port 2049 frame to server.company.com

```
host> echo Starting NFS Mount now --marker | nc -4 -w 1 server.company.com 2049
C:\Temp> echo Starting NFS Mount now --marker | ncat -4 -w 1 server.company.com 2049
For Windows, install the open source ncat utility http://www.insecure.org, part of the Nmap distribution
```

Send a UDP port 666 frame to server.company.com

```
host> echo Starting app now --marker | nc -4 -w 1 -u server.company.com 666
C:\Temp> echo Starting app now --marker | ncat -4 -w 1 -u server.company.com 666
```

Create a file, the name of the file will appear in Wireshark's Summary screen

```
host> touch /mnt/whatever/slowness-starting-now--marker.txt
C:\Temp> copy /y nul z:slowness-starting-now--marker.txt
```

Drop the message into /var/log/syslog on loghost

```
host> logger -n loghost.company.com Slowness starting now --marker
C:\Temp> logger -n loghost.company.com Slowness starting now --marker
For Windows, install the freeware logger utility http://www.monitorware.com/logger
```

Drop the message into the Web server's logs

```
host> wget http://www.company.com/slowness-starting-now--marker.html
C:\Temp> wget http://www.company.com/slowness-starting-now--marker.html
For Windows, install the open source GNU wget utility
```

Drop the message into database server logs

```
SELECT name_last, name_first FROM name_table WHERE name_last ILIKE 'slowness starting now';
```

CLI Ping

In a pinch, you can use ping, manually maintaining a written table associating ping packet length to message:

```
host> ping -n 1 -l 101 server.company.com
```

```
host> ping -n 1 -l 102 server.company.com
```

```
host> ping -n 1 -l 103 server.company.com
```

Ping Packet Length

101 bytes

102 bytes

103 bytes

Event

Mounting file system

Starting application

Slowness beginning now

Or, depending on your filters, ping a fake host ... the ping won't show up in the trace, but the failed DNS query will:

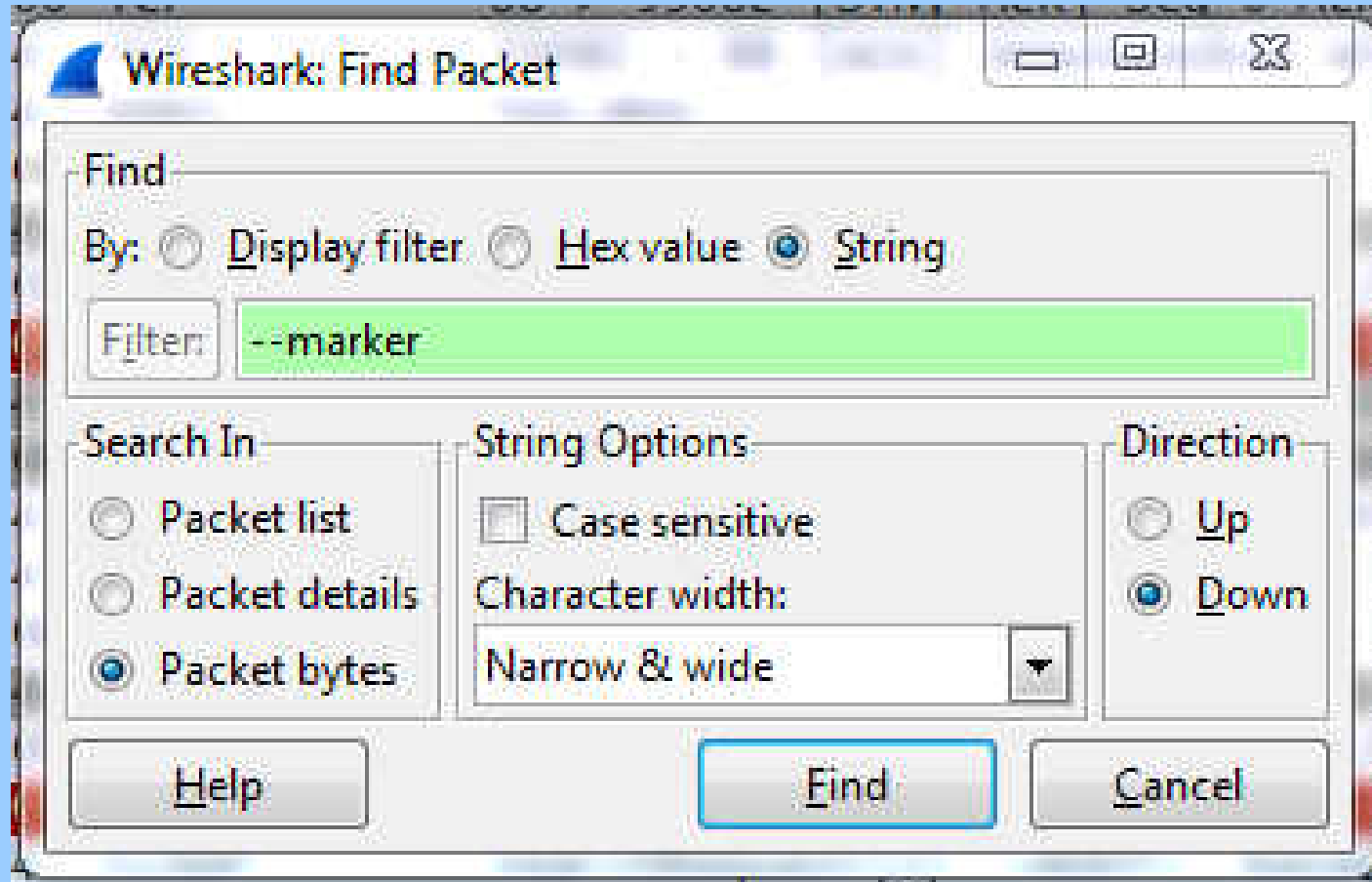
```
host > ping www.slowness-starting-now--marker.com
```

```
C:\Temp> ping www.slowness-starting-now--marker.com
```

How to find these markers?

Once you've opened the trace, how do you find these markers?

Edit menu ... Find Packet



I append the same string to all my markers ... that way I can search through the trace and find them all, without having to remember unique strings for each marker

Questions about Markers?

Questions up to this point ...

So Is that It?

Inserting markers into trace files is a great start and is the RPR technique which I employ most frequently

But there's more ...

Into ProcMon

Time of Day	Process Name	PID	Operation	Path	Result	Detail
8:18:33.6847425 AM	svchost.exe	648	RegQueryKey	HKLM\System\CurrentControlSet\Control\DeviceClasses	SUCCESS	Query: HandleTags. HandleTags: 0x0
8:18:33.6847540 AM	svchost.exe	648	RegOpenKey	HKLM\System\CurrentControlSet\Control\DeviceClasses\{e2d1ff34-3458-49a9-88da-8e6}	SUCCESS	Desired Access: Query Value
8:18:33.6847667 AM	svchost.exe	648	RegQueryKey	HKLM\System\CurrentControlSet\Control\DeviceClasses\{e2d1ff34-3458-49a9-88da-8e6}	SUCCESS	Query: HandleTags. HandleTags: 0x0
8:18:33.6847757 AM	svchost.exe	648	RegOpenKey	HKLM\System\CurrentControlSet\Control\DeviceClasses\{e2d1ff34-3458-49a9-88da-8e6}	NAME NOT FOUND	Desired Access: Query Value
8:18:33.6848093 AM	svchost.exe	648	RegCloseKey	HKLM\System\CurrentControlSet\Control\DeviceClasses\{e2d1ff34-3458-49a9-88da-8e6}	SUCCESS	
8:18:34.1896044 AM	cmd.exe	5940	CreateFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	Desired Access: Read Attributes, Dispo
8:18:34.1896209 AM	cmd.exe	5940	QueryBasicInformationFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	CreationTime: 5/11/2014 8:17:26 AM, Las
8:18:34.1896287 AM	cmd.exe	5940	CloseFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	
8:18:34.1897624 AM	cmd.exe	5940	CreateFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	Desired Access: Read Attributes, Dispo
8:18:34.1897654 AM	cmd.exe	5940	QueryBasicInformationFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	CreationTime: 5/11/2014 8:17:26 AM, Las
8:18:34.1898221 AM	cmd.exe	5940	CloseFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	
8:18:34.1899191 AM	cmd.exe	5940	CreateFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	Desired Access: Generic Write, Read Attr
8:18:34.1900308 AM	cmd.exe	5940	SetEndOfFileInformationFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	EndOfFile: 0
8:18:34.1900958 AM	cmd.exe	5940	SetAllocationInformationFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	AllocationSize: 0
8:18:34.1901263 AM	cmd.exe	5940	CloseFile	C:\Temp\skewing-clock-on-admt01-marker.txt	SUCCESS	
8:18:34.2944822 AM	Explorer EXE	1636	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: Name
8:18:34.2945036 AM	Explorer EXE	1636	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: HandleTags. HandleTags: 0x0
8:18:34.2945164 AM	Explorer EXE	1636	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: HandleTags. HandleTags: 0x0
8:18:34.2945344 AM	Explorer EXE	1636	RegOpenKey	HKCU\Software\Classes\Applications\cmd.exe	NAME NOT FOUND	Desired Access: Read

5940	Create File	C:\Temp\skewing-clock-on-admt01-marker.txt
5940	Query Basic Information File	C:\Temp\skewing-clock-on-admt01-marker.txt
5940	Close File	C:\Temp\skewing-clock-on-admt01-marker.txt
5940	Create File	C:\Temp\skewing-clock-on-admt01-marker.txt
5940	Query Basic Information File	C:\Temp\skewing-clock-on-admt01-marker.txt

Into strace

```
gnat> strace -p 12345 -f -tt -s 256
[...]
```

08:35:19.764185 mprotect(0x7f5e8f841000, 4096, PROT_READ) = 0

08:35:19.764261 mmap(0x7f5e8f82e000, 65819) = 0

08:35:19.764374 set_tid_address(0x7f5e8f82b9d0) = 10506

08:35:19.764498 set_robust_list(0x7f5e8f82b9e0, 0x18) = 0

08:35:19.764612 futex(0x7fff659d2e9c, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 1, NULL

08:35:19.764820 rt_sigaction(SIGRTMIN, {0x7f5e8ee40750, [], SA_RESTORE|SA_SIGINFO, 0x7f5e8ee49cb0}, NULL

08:35:19.765027 rt_sigaction(SIGRT_1, {0x7f5e8ee407e0, [], SA_RESTORE|SA_RESTART|SA_SIGINFO,

08:35:19.765210 rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

08:35:19.765340 getrlimit(RLIMIT_STACK, {rlim_cur=8192*1024, rlim_max=RLIM_INFINITY}) = 0

08:35:19.766014 brk(0) = 0x1a2d000

08:35:19.766073 brk(0x1a4e000) = 0x1a4e000

08:35:19.766143 open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3

08:35:19.766220 fstat(3, {st_mode=S_IFREG|0644, st_size=7220736, ...}) = 0

08:35:19.766287 mmap(NULL, 7220736, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f5e8e757000

08:35:19.766844 close(3) = 0

08:35:19.766966 open("Skewing-clock-on-admt01--marker", O_WRONLY|O_CREAT|O_NOCTTY|O_NONBLOCK, 0666) = 3

08:35:19.767746 dup2(3, 0) = 0

08:35:19.768110 close(3) = 0

08:35:19.768172 dup2(0, 0) = 0

08:35:19.768227 utimensat(0, NULL, NULL, 0) = 0

```
[...]
```


Let's Back Out A Bit

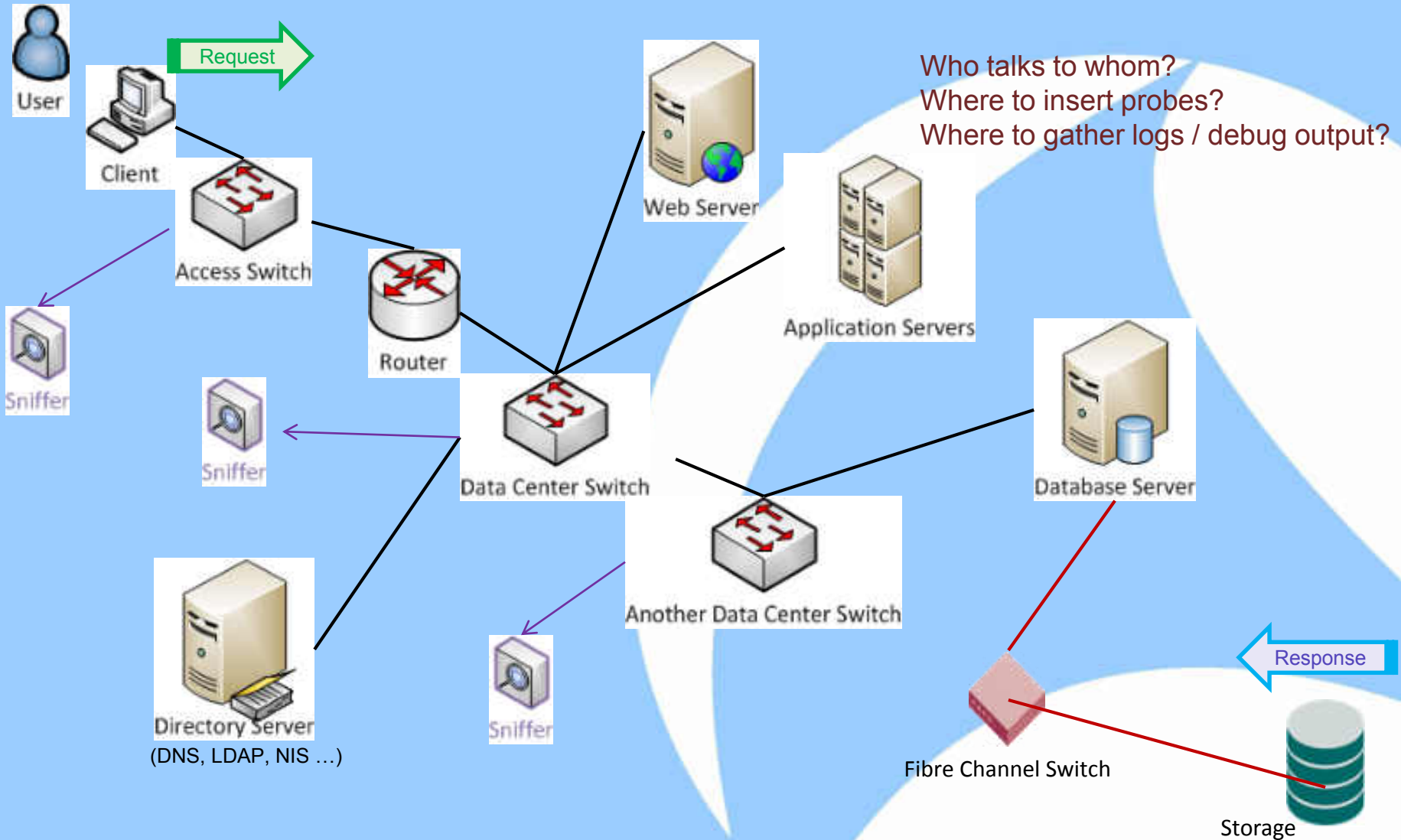
I chose to present this topic by diving immediately into the details of how to use markers to guide our attention during pcap analysis.

And have now expanded the use of markers to include guiding our attention during log analysis.

And guiding our attention during analysis of the output of trouble-shooting tools like ProcMon and strace.

I would like to pull back from examining the leaves to examine the entire tree.

Diagnostic Capture Plan



Definitive Diagnostic Data

Often, we solve problems through an intuitive approach, gathering a pcap here, a log file there, running an experiment, and at last identifying the root cause.

RPR suggests that for a certain class of big, hairy problem, our usual approaches not only cost a lot in terms of time *but may never converge on the answer*.

RPR proposes that for problems of sufficient complexity, particularly intermittent ones, we take the time to:

1. Instrument the entire path of the troubled transaction
2. Validate the data collection tools by sending a sample transaction end-to-end
UPDATE salary_table SET annual=100000 WHERE name="Mickey Mouse"
3. Lie in wait for the next occurrence of the problem, sending markers when it happens
4. Analyze the resulting 'video' of the troubled transaction, using the end-to-end view captured in D³, to definitively finger the portion of the path causing the problem

It's a judgment call on when it is worth investing this level of effort.

Judgment Call

It's a judgment call on when it is worth investing this level of effort.

Myself, I tend to explore for a while using my intuitive approach, before finally accepting the need to bring out the power tool: *Definitive Diagnostic Data*

In fact, I tend to tell myself:

Self *This time, I'll solve the problem using some clever shortcut, and I won't need to spend all that time setting up a formal Diagnostic Capture Plan*

[Many hours or days later]

Self *Wish you'd started with D^3 , don't you? Thought you could get away with a shortcut, didn't you?*

I like to think that optimism is a desirable character trait ... 😊

The Fantasy ...

Ideally, you're sitting next to the end-user's computer, reading {insert your favorite book here} when she says "There, see, it's hanging right now!" And you ask her to double-click on the desktop icon which fires off your marker:

```
UPDATE salary_table SET annual=100000 WHERE name="Mickey Mouse"
```

Then, you walk back to your desk and double-click the icon which automatically logs into each sniffer, copies the latest pcap (neatly named), logs into each server, copies the last 15 minutes of logs ... ProcMon ... strace ... and collects all these data files into a folder named Analyze-Me.

On your four 30" monitors, you open all the files, organized from left to right reflecting the Diagnostic Capture Plan diagram, and you search in each one for 'Mickey Mouse', lining them all up. Immediately, you can see that the Database Server took two minutes to return the OK symbol, and during that time, the DBM was logging messages like "Table *salary* locked exclusive, waiting", you walk over to the database manager, describe what you see, he smacks his forehead and says "Duh! Of course! I've been meaning to fix that for months ... [type, type] ... OK try it now, I bet it works" ...

and ... you earn your paycheck that day.

The Reality ...

This is not as easy as it looks. –The Man in Black

- The application encrypts its traffic over the wire, so you can't see *Mickey Mouse* in the pcaps
- So when the user clicks the icon, she's sending a custom ping packet through the front-half of the transaction, while you attempt, as simultaneously as you can, to send another custom ping packet through the back-half of the transaction – they won't be lined up perfectly, but hey, the best you can do
- Simultaneously, you give your colleagues a shout and they insert custom log messages into the logs of various applications along the way: *Mickey Mouse is Here ...* again not perfectly lined up

And then you discover:

- One of the sniffers was hung
- You fat-fingered the marker insertion at one of the steps
- You forgot to copy the logs off a particular server and by the time you notice, they have been overwritten

More tips

Automate as much typing as possible:

Batch files (PowerShell, bash, Perl, Python, whatever)

So, instead of typing:

```
echo Starting NFS Mount now --marker | ncat -4 -w 1 server.company.com 2049
```

Write shell scripts to do the same: *ins-start-marker* and *ins-end-marker*

The checklist is your friend: [The Checklist Manifesto](#) by Atul Gawande

e.g.

When the pathology starts, we're going to do the following:

Bob double-clicks on 'Pathology-Starting' icon

Sarah types *ins-start-marker* on the Web Server

Jiang types *ins-start-marker* on the Application Server

...

Example of a plan coordinating the efforts of a team during a maintenance window:

<http://www.skendric.com/philosophy/uptime/DaPlan-Hobbes.pdf>

Back Out Even Further

Let's back out and look at the entire forest, or at least at a grove of trees

How Do Techs Fix Issues?

Oh boy, that's a big question. But let's take a stab at answering it. A tech might start asking themselves, or the person reporting the problem, questions similar to the following:

- What makes you think there is an issue?
- What are you expecting that you're not getting?
- Has it ever performed well?
- **What changed recently?** Software or hardware? Load?
- Can it be expressed in terms of latency or run time?
- Does the problem affect **other people or applications?**
- What is the environment? What software and hardware is used? **Versions? Configuration?**
- ...

Most issues get fixed somewhere during the process of asking these questions and uncovering the answers ...

Anti-Patterns

As the issue resists resolution, less skilled techs will start employing less effective approaches.

Street Lamp Method

The student comes across his professor on the Arts Quad at night, down on his hands & knees, staring at the sidewalk. “What are you doing, sir?” “Looking for my car keys”. The student joins the professor but after looking unsuccessfully in widening circles, asks him “Do you recall precisely where you were when you dropped the keys?” “Yes, over there, in the middle of the quad” points the professor, toward the dimly perceived middle of the grassy acre. “Well, why are you looking here?” asks the student. “Because the light is better here” responds the professor.

More formally:

:START

1. List available tools
2. Examine the output of each one, looking for clues
3. Purchase more tools
4. Goto START

Use The Force, Luke

“I know that we are experiencing a broadcast storm ... you should check your {switch | router | firewall | server | client | application | whatever-belongs-to-some-other-group}”

I enjoyed Star Wars ... but it was fiction, not real ... that distinction is hard for human brains to make. --sk

When It Really Hurts

What happens when your technical teams (desktop, server, network, storage, database, application ...) have looked at the issue and are stumped?

Or worse, have tried something, and it didn't help ...

Or even worse ... are now avoiding the issue ...

Your vendor tells you to upgrade to the next biggest model / version ...

Tensions rise, people point fingers ... blame-based language ...

What's next?

Rapid Problem Resolution ®

Advance7 is a consulting outfit which helps customers resolve critical Problems – they put an analyst at your site to coordinate your staff plus vendors to fix the issue, using the RPR methodology. They tend to play in the Fortune 1000 + government space.

RPR is an evidence-based trouble-shooting approach.

Q: Aren't all trouble-shooting methods evidence-based? A: Regrettably, not.

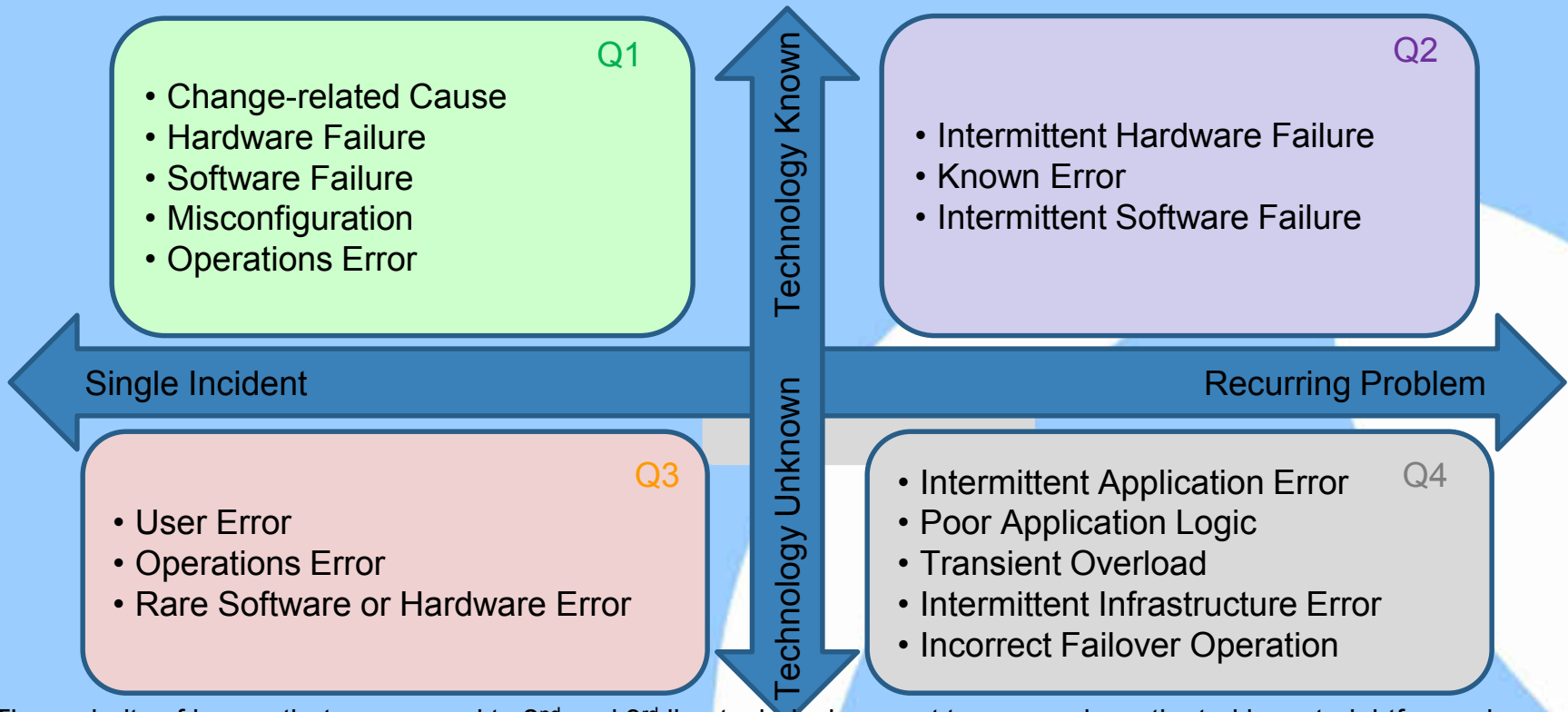
Advance7 designed RPR to work against **Grey Problems**

Most Problems are not Grey ... unless the Problem is Grey, RPR is overkill.

So what are Grey Problems?

The following sides are cribbed from Advance7 -- credit to Paul Offord & his colleagues.

The Grey Problem

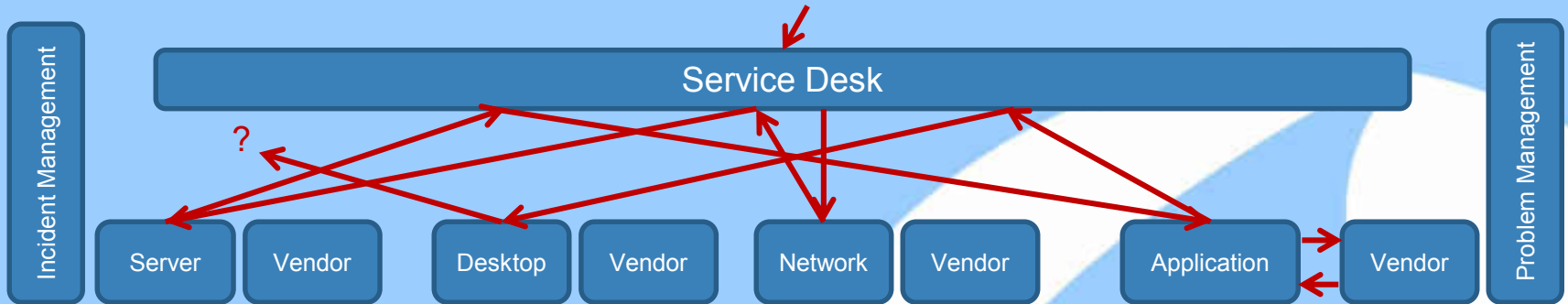


The majority of issues that are passed to 2nd and 3rd line technical support teams are investigated in a straightforward manner. The nature of the issue or an indication from a monitoring system identifies the failing component, and the issue is assigned to the correct technical support team. **Q1: the bulk of support work falls into this area.** **Q2 is harder but tends to be resolved by experienced support staff.** **Q3 are sticky; we tend not to solve these.**

An intermittent response-time or error issue is not so easily handled due to its transient nature. Not only does the cause sneak under the radar of monitoring systems, but investigation often starts after the issue has passed, making it impossible to use many of the tools available. The result is a recurring problem where the causing technology is unknown: **Q4, aka the Grey Problem.** **The Rapid Problem Resolution methodology targets Q4.**

Grey Problem Characteristics

Because the causing technology is unknown, a grey problem will bounce between Technical Support Teams as each in turn produces evidence (often in the form of a health check) to prove that their technology is not to blame.



Typical characteristics of a grey problem

- An ever-growing number of people become involved
- Long meetings to discuss what might be the cause
- Support people shy away from becoming involved
- Repeated changes with no clear reason or objective
- Technical Support Teams hire their vendors to perform health checks

Consequences of grey problems

- An ever growing backlog of problems
- A fog that hinders the investigation of other, more urgent problems
- A growing pool of problems that escalate into Major Incidents as patterns of use and business priorities change
- Wasted IT budget as money is spent on poorly targeted upgrades
- Barriers to integration due to concerns about the stability of component systems
- Loss of confidence and satisfaction with the IT department
- Pressure to outsource IT services
- Reduced customer satisfaction
- Higher costs as the business adjusts to accommodate the problem
- Higher IT staffing costs

RPR Roles & Responsibilities

Who	What
Facilitator (often a Problem Manager)	<ul style="list-style-type: none">• Accountable for<ul style="list-style-type: none">○ Owns the RCA○ Acquire resources○ Use and execute the methodology○ Communicate within the team○ Report & escalate to leadership○ Schedule meetings
Problem Analyst (often a senior tech)	<ul style="list-style-type: none">• Responsible for<ul style="list-style-type: none">○ Unify & synthesize information from SMEs○ Keep team on track technically○ Breadth & depth
Subject Matter Experts	<ul style="list-style-type: none">• Responsible for<ul style="list-style-type: none">○ Strong fundamental knowledge of area○ Facilitating access○ Capturing data○ Analyzing
SME Desirable Characteristics	<ul style="list-style-type: none">• Skills / Predilections<ul style="list-style-type: none">○ Problem solving skills○ Inquiring mind – passion for understanding how things work○ Determination & stamina – pursuing a tough problem can be wearing○ T-shaped – broad background in IT with specialization in one or two particular areas

Simplified RPR

1. Understand the Symptoms
2. Choose One Symptom
3. Understand the Symptom Environment

4. Design Diagnostic Capture Plan
5. Capture *Definitive Diagnostic Data*
6. Analyze Captured Data

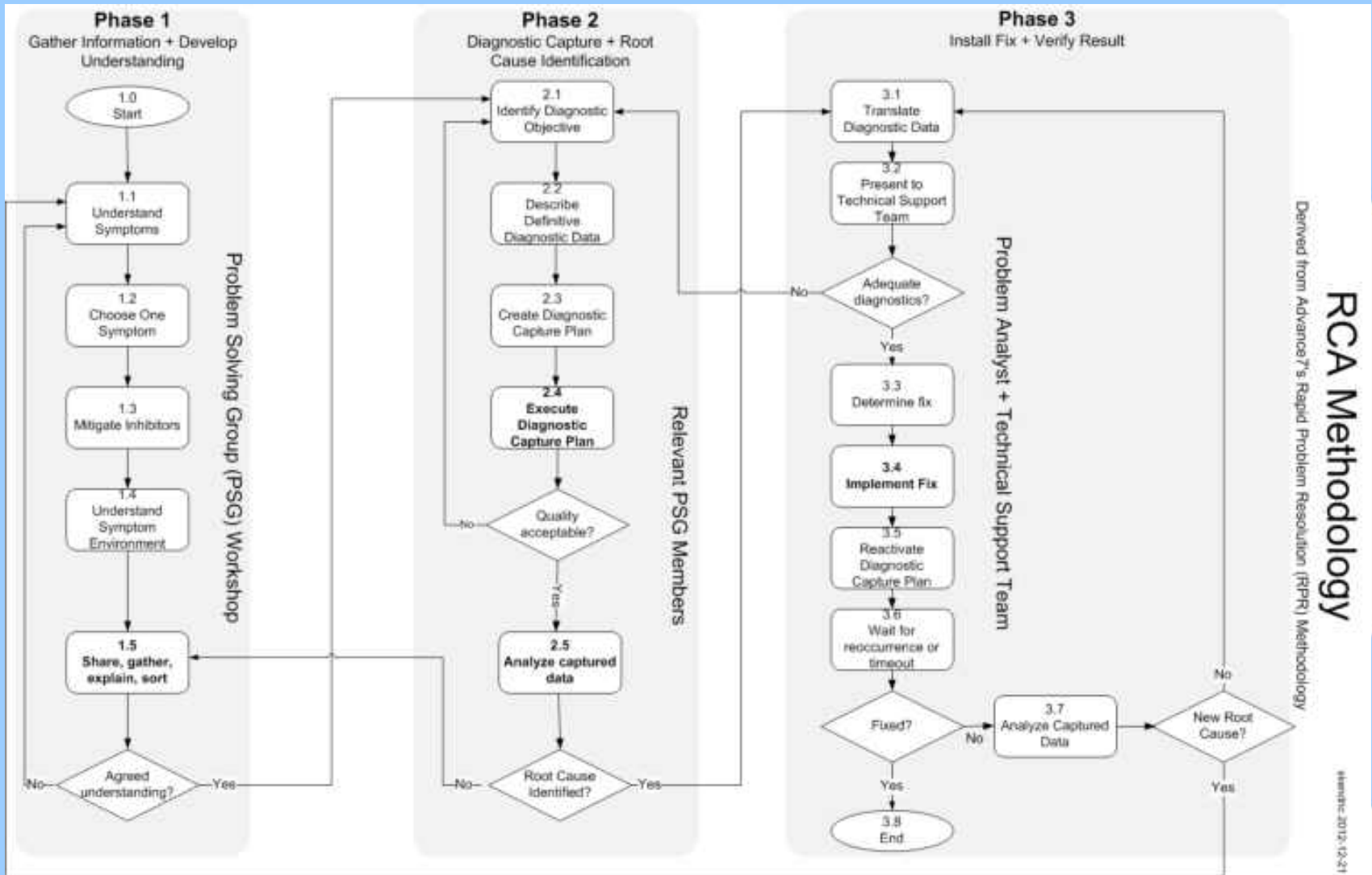
7. Identify Fix
8. Implement Fix
9. Verify Fix

Phase 1

Phase 2

Phase 3

The Full RPR Methodology



See the [Rapid Problem Resolution](#) book and [Advance7's Web site](#) for more information

Summary

1. Instrument the entire path of the troubled transaction

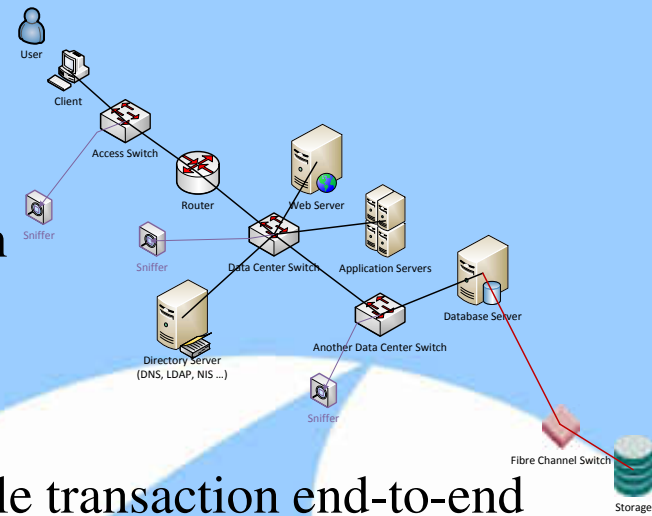
2. Validate the data collection tools by sending a sample transaction end-to-end
Increase Mickey Mouse's salary to 100,000 --marker

3. Lie in wait for the next occurrence of the problem.

Ideally, the troubled transaction shows up in a recognizable way at each probe: *“I was updating the customer's address from 100 Main St to 200 Broad St when my application froze”* – search for Broad St.

But you may need to manually insert markers into the data stream to build a richer trail of bread crumbs

4. Analyze the resulting ‘video’ of the troubled transaction, using the end-to-end view captured in D³, to definitively finger the portion of the path causing the problem



Thank you!

On-Line Resources

[Rapid Problem Resolution](#) by Paul Offord

LinkedIn [Protocol Analysis & Troubleshooting Group](#)

Old Comm Guy <http://www.lovemytool.com>

Trouble-shooting & Training Outfits

Based Here (will travel for \$\$)

James Baxter <http://www.packetiq.com>

Daytona Beach, FL

Tony Fortunato <http://www.thetechfirm.com>

Toronto, Canada

Chris Greer <http://www.packetpioneer.com>

Central/South America

Paul Offord <http://www.advance7.com>

London (international)

Mike Pennacchi <http://www.nps-llc.com>

Seattle, WA

Ray Tompkins <http://www.gearbit.com>

Austin, TX

...

Conferences

Sharkfest <http://www.sharkfest.org>

San Francisco, CA

Follow-up

stuart.kendrick.sea {at} gee mail dot com

This deck visible at <http://www.skendric.com/seminar>