Fun With Traces – Chapter 3

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

Stuart Kendrick
Systems Engineer
Allen Institute for Brain Science

The Concept

Fun With Traces

I developed this seminar as a day-long Hands-On Lab, in which we practice Wireshark techniques while analyzing real-world case studies.

Today, we have 75 minutes. Hmm. So, we'll shrink the solo practice time down to ~5 minutes per trace. And won't reach all the traces. But I still predict fun!

I try to slip a lot of lessons into this format, from refining the Problem Statement to diagramming examples to leveraging your Problem Management process for communicating risk.

I promote interactivity: please interrupt, contribute, heckle as you see fit. Then again, if you prefer to sit back, watch, and listen, you are welcome to do that also.

I predict that you know ways to analyze these cases faster/better than I did … please share your techniques, and I'll demo them for all to see.

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

Traces

- Grab a USB stick from the basket up front
- Or download from http://www.skendric.com/Sharkfest2014

This deck available at http://www.skendric.com/seminar/

Me

Multi-disciplinary IT trouble-shooter / Root Cause Analysis
http://www.skendric.com

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

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

Geeky Highlights

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

Case Studies

Case 1	The Network is Slow	We Need a Bigger Boat
Case 2	Intermittent File Copy Failures	The Network Sucks
Case 3	Router Misses Pings	The Moles Rule
Case 4	The Internet is Slow	Must Be The Firewall
Case 5	The Web Server is Slow	Mouse Brains
Case 6	Intermittent Citrix Disconnects	Xenophobia

Case 1
The Network is Slow

We Need a Bigger Boat

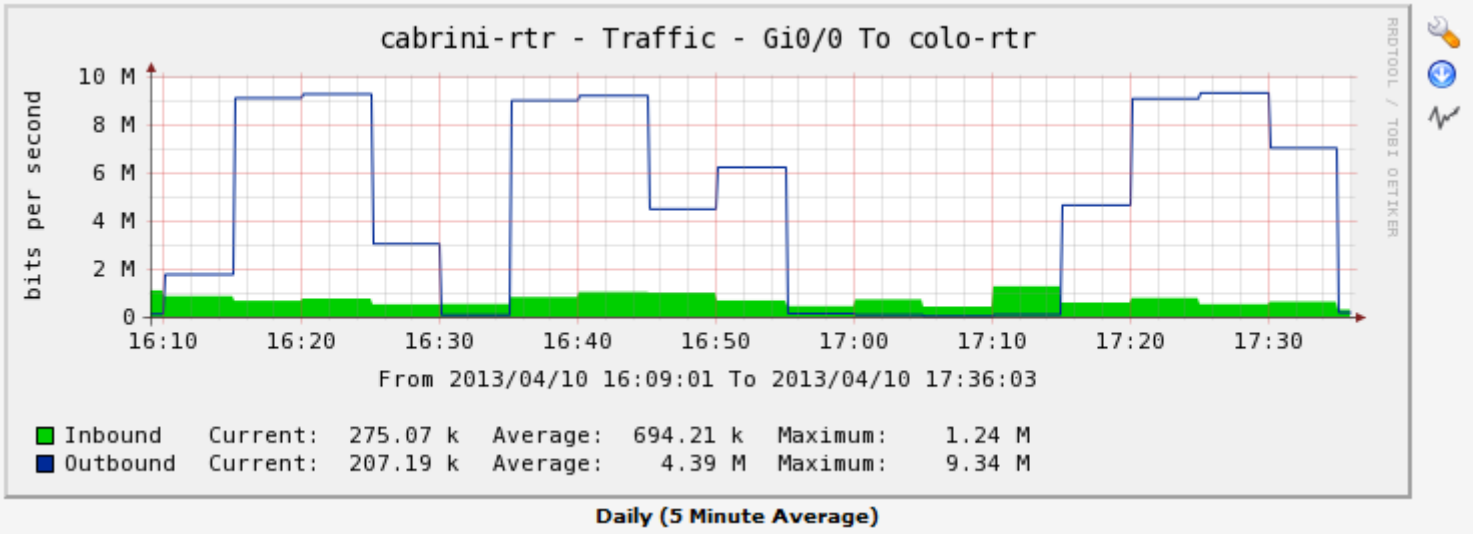
Case 1: Background

A remote office links back to the main campus via a 10MB/s Transparent LAN Service (TLS), a distance of several miles as the crow flies. Users complain of poor performance and want a bigger WAN pipe – they suggest that 100MB/s sounds like the next step

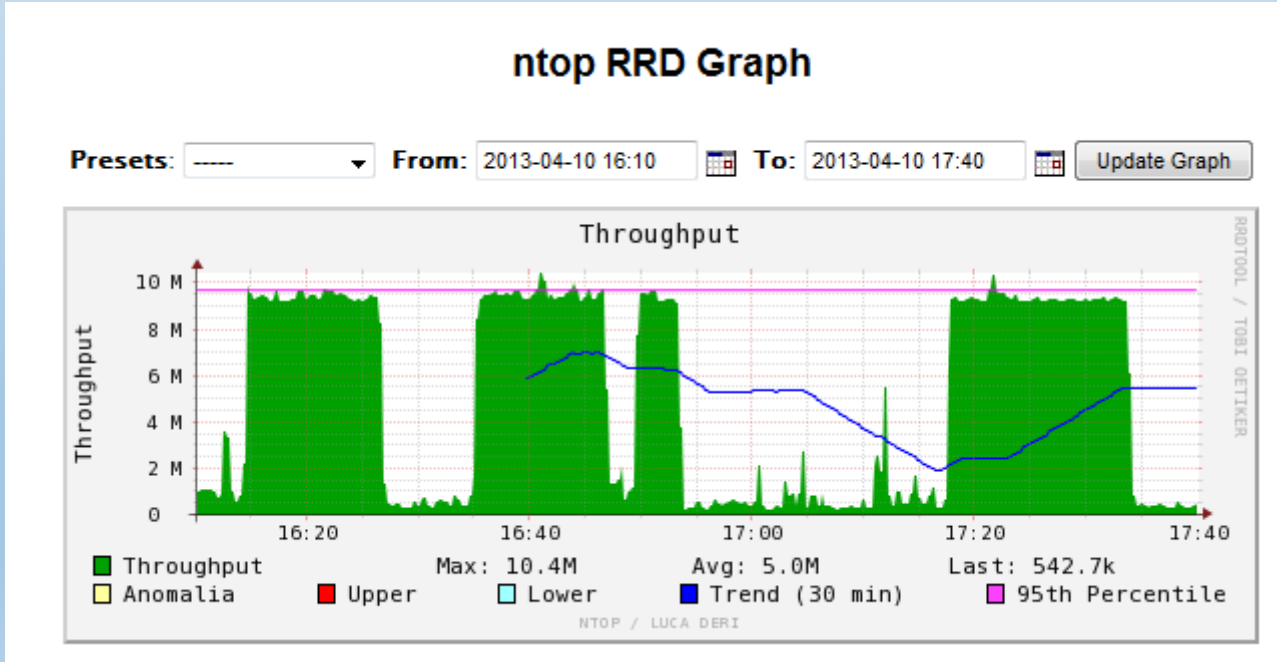
The current monthly cost is straining the budget, jumping to 100Mb/s would double or triple the cost and require signing a long-term contract … management wants evidence that taking this step will improve the end-user experience

You are asked to estimate the effect on the end-user experience, were the TLS to be upgraded

Cacti


cabrini-rtr - Traffic - Gi0/0 To colo-rtr
bits per second
10 M
8 M
6 M
4 M
2 M
0
16:10
16:20
16:30
16:40
16:50
17:00
17:10
17:20
17:30
From 2013/04/10 16:09:01 To 2013/04/10 17:36:03
Inbound Current: 275.07 k Average: 694.21 k Maximum: 1.24 M
Outbound Current: 207.19 k Average: 4.39 M Maximum: 9.34 M
Daily (5 Minute Average)
RRDTOOL / TOBI OETIKER

nTop


ntop RRD Graph
Presets: -----
From: 2013-04-10 16:10
To: 2013-04-10 17:40
Update Graph
Throughput
Throughput
10 M
8 M
6 M
4 M
2 M
0
16:20
16:40
17:00
17:20
17:40
Throughput Max: 10.4M Avg: 5.0M Last: 542.7k
Anomalia Upper Lower Trend (30 min) 95th Percentile
NTOP / LUCA DERI
RRDTOOL / TOBI OETIKER

Case 1: Problem Statement

For the purposes of this task, we're not analyzing a problem – we're not trying to figure out why end-users experience slow performance for example – so I don't have a classic Problem Statement to offer

Instead, we're attempting to perform an *Application Assessment* – given a hypothetical change in the environment, how would a given application behave? Colloquially, I call this *Predicting the Future* … in my experience, a generally fruitless endeavour

Prediction is very difficult, especially if it's about the future – Niels Bohr
Predicting is hard, particularly the future – Yogi Berra

Nevertheless, no need to throw our hands up in despair and run away. Instead, we can at least gather clues which management can use to inform this decision

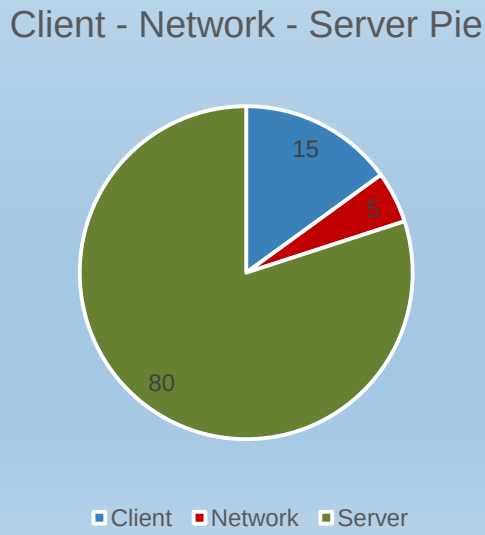
Ideally, we would tackle this task from various angles, typically using simulation for each of the applications in use. For our purposes today, we'll pick one application (file copy) and use packet traces to draw the Client-Network-Server Pie. Is this the best way to inform management's choice? No, but this is a class about trace analysis, so here we go

Case 1: Traces

We have two traces, one taken from the remote office (the sniffer *Caveman* plugged into a mini-hub shared by a user's workstation), the other taken from the main campus (the sniffer *Flim* plugged into a SPAN port on the switch at the main campus end of the TLS)

Effectively, for our purposes today, we have traces taken from both ends of the TLS, encapsulating about a minute of a file copy

Using these two traces, we will *Draw the CNS Pie*, estimating how much time the Client contributes to this file copy, how much time the Network contributes, and how much time the Server contributes



Given the pie, we will then offer insights into what effect changing a component might have. For example, in the Pie on this page, we can see that shrinking the Network portion would not have much effect on overall transaction time – the place where we really want to put our attention is on the Server

Case 1: Making Pie

There are lots of ways to do this. I review a few in a paper called Making Client / Network / Server Pie http://www.skendric.com/app/

These techniques are all fragile – there are many ways in which they can go south and deliver misleading results

High-end applications, e.g *OpNet* et al, attempt to robustify this process – I've never used them, but I hear that they do a good job; though unsurprisingly, they can make mistakes also

There is no silver bullet

In the approach I sketch here, the basic idea is to Sum all the DeltaTs between frames emitted by the Client (in the Client-side trace), Sum all the DeltaTs between frames emitted by the Server (in the Server-side trace), subtract those two from the total trace time, and use the resulting three numbers to Draw the Pie.

How might we do that?

Case 1: Client-Side Sum DeltaT

(1) Produce a file which contains traffic only between Client & Server for a single conversation

```
cd Events\2013-04-10\1650
tshark –r caveman.pcap –w caveman-client–and-server-only.pcap –Y
“ip.addr==10.111.42.34 and tcp.port==445”
```

(2) On the Client-side trace, ask *tshark* to calculate how much time the Client spent turning around frames

```
tshark -r caveman-client-and-server-only.pcap –qz
“io,stat,0,SUM(frame.time_delta)frame.time_delta and ip.src==140.107.203.6”
```

```
================================================================================
| IO Statistics                                                                |
|                                                                              |
| Interval size: 44.1 secs (dur)                                               |
| Col 1: Frames and bytes                                                      |
|     2: SUM(frame.time_delta)frame.time_delta and ip.src==140.107.203.6       |
|------------------------------------------------------------------------------|
|              |1                  |2          |                               |
| Interval     | Frames |   Bytes  |    SUM    |                               |
|-----------------------------------------------|                               |
|  0.0 <> 44.1 |  50071 | 50751753 |  9.529210 |                               |
================================================================================
```

Case 1: Server-Side Sum DeltaT

(3) Produce files which contain traffic only between Client & Server for a single conversation

```
cd \events\2013-04-10\1650
tshark –r flim.pcap –w flim-client-server-only.pcap –Y “ip.addr==140.107.203.6
and tcp.port==445”
```

(4) On the Server-side trace, ask *tshark* to calculate how much time the Server spent turning around frames

```
tshark -r flim-client-server-only.pcap -qz
“io,stat,0,SUM(frame.time_delta)frame.time_delta and ip.src==10.111.42.34”
```

```
===================================================================
| IO Statistics                                                   |
|                                                                 |
| Interval size: 44.1 secs (dur)                                  |
| Col 1: Frames and bytes                                         |
|     2: SUM(frame.time_delta)frame.time_delta and ip.dst==140.107.203.6 |
|-----------------------------------------------------------------|
|              |1                  |2         |                   |
| Interval     | Frames |   Bytes  |    SUM   |                   |
|--------------------------------------------|                   |
|  0.0 <> 44.1 |  50069 | 50551304 | 3.001946 |                   |
===================================================================
```

Case 1: Total Time

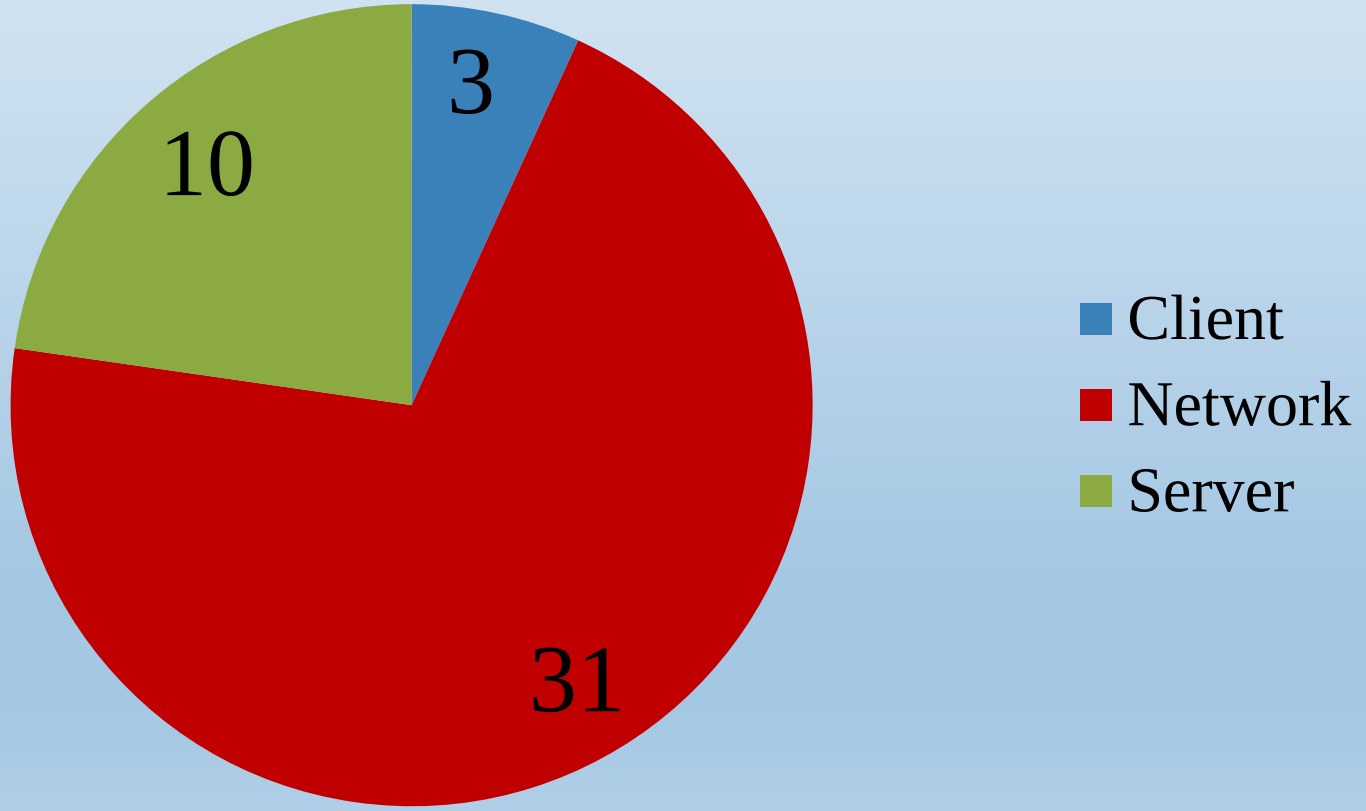
(5) How much time total do these traces cover?

```
C:\Temp> capinfos caveman.pcapng
File name:           caveman.pcapng
File type:           Wireshark/... - pcapng
File encapsulation:  Ethernet
Packet size limit:   file hdr: (not set)
Number of packets:   50 k
File size:           52 MB
Data size:           50 MB
Capture duration:    44 seconds
Start time:          Wed Apr 10 16:48:59 2013
End time:            Wed Apr 10 16:49:44 2013
Data byte rate:      1149 kBps
Data bit rate:       9197 kbps
Average packet size: 1013.07 bytes
Average packet rate: 1134 packets/sec
SHA1:                e6461c3ef2cb009beb048706e89b8248f587b228
RIPEMD160:           da28a8dac1d756a6f439d443636d96a0319d3254
MD5:                 bf6d59e8cd1d28e10fbef47718012980
Strict time order:   True
```

Case 1: Draw the Pie

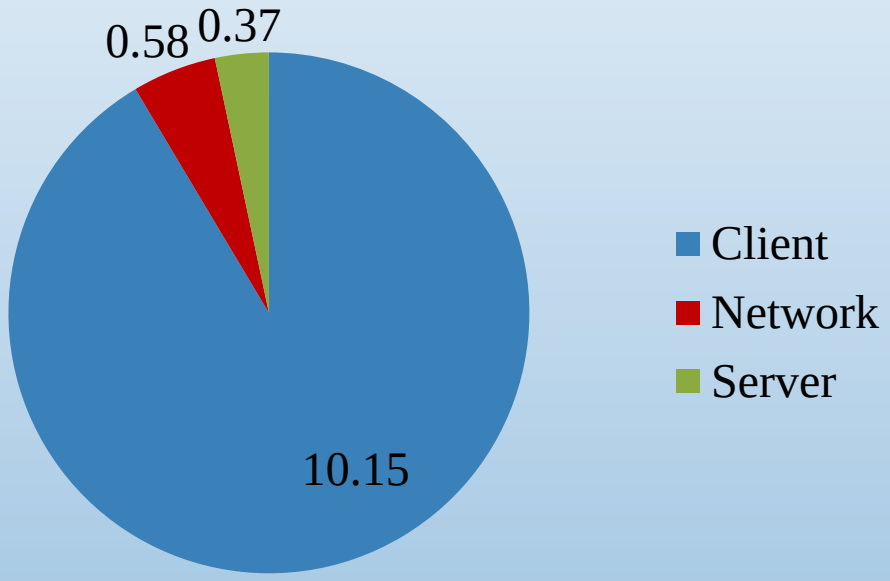
(6) OK, so the Client consumed ~10s, the Server consumed ~3s: how much time did the Network consume?

```
44s - 10s - 3s = 31s
```



Case 1: Draw More Pies

If we repeat this for other applications, we find, for example, that for browsing Internet Web sites, the pie looked liked this. Hard to believe? Turns out the Client was struggling with not enough memory.



The Outlook pie looked rather like the one above, while browsing Internal Web sites split the time more evenly between the three components.

This approach is fragile – what kinds of pathology can defeat it?

Case 1: Outlook

Write a little script to send your test Exchange account a thousand messages

```
#!c:\perl64\bin\perl

# Load modules
use strict;
use warnings FATAL => 'all';
use feature 'say';
use DateTime;
use English;
use Mail::Outlook;
use Time::HiRes qw(time);

# Declare variables
my $datestamp;
my $dt;
my $diff;
my $end;
my $folder;
my $logfile;
my $outlook;
my $start;
my $text;

# Define variables
$OUTPUT_AUTOFLUSH = 1;
$logfile = 'create-messages.log';

# Create Outlook object
$outlook = Mail::Outlook->new('Inbox');

# Open log file, create Date Object
open my $log, '>>', $logfile or die
"Cannot open $logfile: $!";
say 'Sending messages';
$dt = DateTime->now(time_zone =>
'local');
$start = time;

# Send a thousand messages
for (my $n = 1; $n <= 1000; $n++) {
 my $message;
 print "$n ";
 $message = $outlook->create();
 $message->To('test@widgets.com');
 $message->Subject("Test Msg $n");
 $message->Body('This is a test
message');
 $message->send;
}

# Clean-up
$end = time;
$diff = $end - $start;
$datestamp = $dt->ymd . ' ' . $dt->hms;
printf {$log} "$datestamp, $diff\n";
say('');
close $log or warn "Cannot close
$logfile: $!";
```

Case 1: Outlook

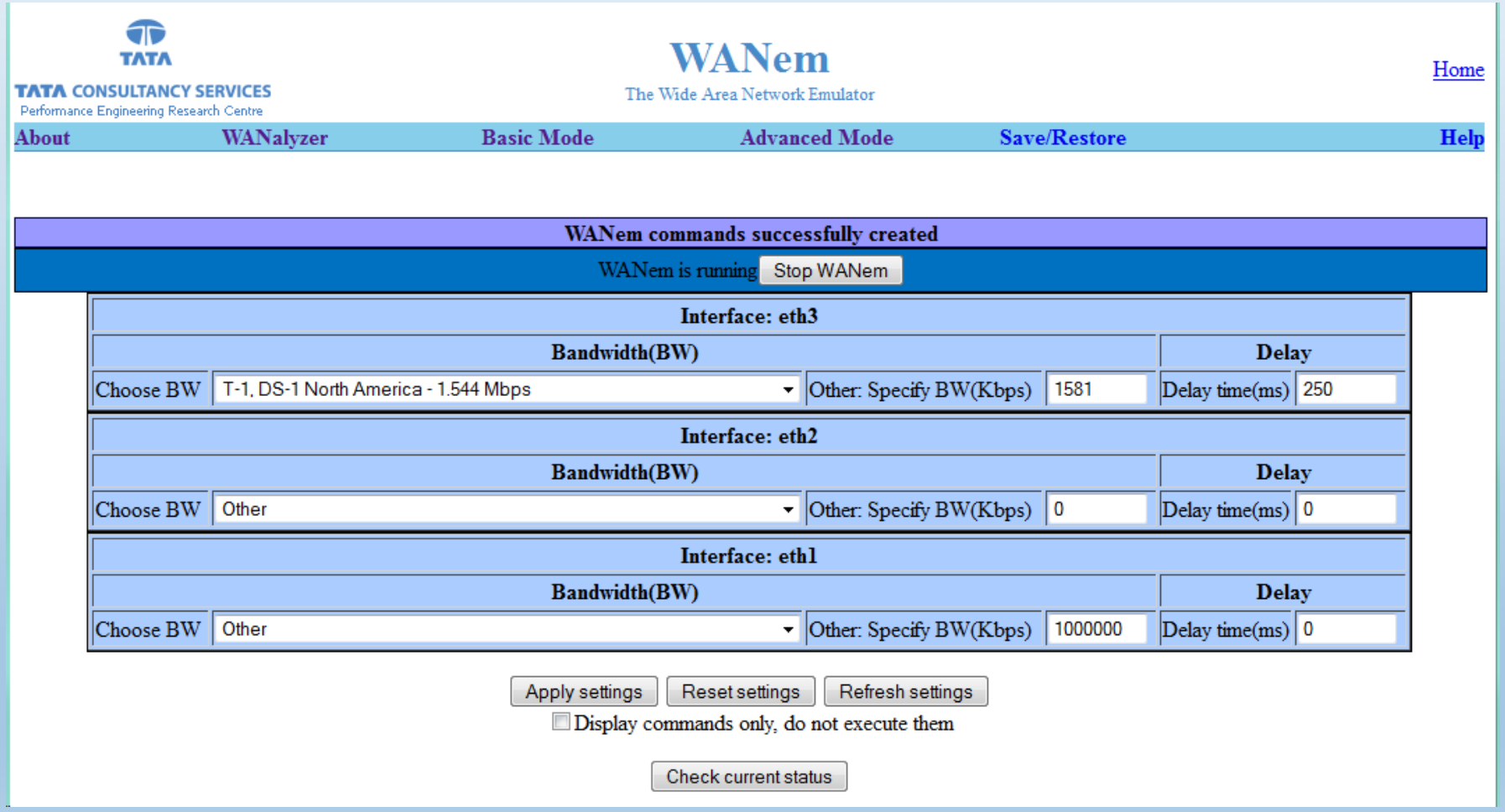
Write a little script to read messages

```
#!c:\perl64\bin\perl

# Send test account a thousand messages

# Load modules
use strict;
use warnings FATAL => 'all';
use feature 'say';
use DateTime;
use English;
use Mail::Outlook;
use Time::HiRes qw(time);

# Declare variables
my $datestamp;
my $dt;
my $diff;
my $end;
my $folder;
my $logfile;
my $outlook;
my $start;
my $text;

# Define variables
$OUTPUT_AUTOFLUSH = 1;
$logfile = 'read-messages.log';

# Create Outlook object
$outlook = Mail::Outlook->new('Inbox');

# Open log file, create Date Object
open my $log, '>>', $logfile or die "Cannot
open $logfile: $!";
say 'Sending messages';
$dt = DateTime->now(time_zone => 'local');
$start = time;
```

```
# Run forever, reading mail
while (1) {
  # Open log file
  open my $log, '>>', $logfile or die "Cannot
open $logfile: $!";
  $dt = DateTime->now(time_zone => 'local');
  say "Reading messages at $dt";
  $start = time;
  $folder  = $outlook->folder('Inbox');

  # Read messages
  for (my $n = 1; $n <= 1000; $n++) {
    my (@list, $message, $text);
    print "$n ";
    $message = $folder->first();
    $text = $message->From();
    $text = $message->To();
    $text = $message->Cc();
    $text = $message->Bcc();
    $text = $message->Subject();
    $text = $message->Body();
    @list = $message->Attach();
    $message->display;
  }
  $end = time;
  $diff = $end - $start;
  $datestamp = $dt->ymd . ' ' . $dt->hms;
  printf {$log} "$datestamp, $diff\n";
  say("\n");
  close $log or warn "Cannot close $logfile:
$!";
  say 'Sleeping';
  sleep 60;
}
```

Case 1: Simulation

Acquire a WAN emulator – e.g. Tata Consulting's WANem or Google's WANBridge – install it on a PC with two NICs



Case 1: Gather Data

Now run your mail reading script for a few days, gathering data, which looks like this:

```
Outlook-Read-over-10Mb-WAN-Circuit.txt
2013-03-12 12:54:14, 17.1627798080444
2013-03-12 12:55:31, 19.2365529537201
2013-03-12 12:56:50, 20.3969390392303
2013-03-12 12:58:10, 16.1613931655884
2013-03-12 12:59:27, 17.4561769962311
2013-03-12 13:00:44, 20.4271380901337
2013-03-12 13:02:04, 20.6425361633301
…

Outlook-Read-over-100Mb-WAN-Circuit.txt
2013-03-13 13:56:33, 23.4620988368988
2013-03-13 13:57:57, 23.7126960754395
2013-03-13 13:59:21, 23.5712969303131
2013-03-13 14:00:44, 25.5368728637695
2013-03-13 14:02:10, 24.0860910415649
2013-03-13 14:03:34, 22.7289090156555
2013-03-13 14:04:57, 24.6944839954376
…
```

Case 1: Crunch Numbers

Perform basic stats on the results

	10Mb/s	**100Mb/s**
Sample Count	6846	789
Average	34	23
Median	23	23
Max	127	29
Min	14	15
StdDev	885	1

Would upgrading to a 100Mb/s TLS improve the responsiveness of Outlook?

Case 1: Do the same for Web browsing

Write a script to grab a Web page (~2MB in size):

```
:START
timethis "wget -Q2m --quiet --recursive --convert-links --page-requisites --restrict-file-names=windows --domains widgets.com --no-parent www.widgets.com > junk 2>&1" >> get-widgets-home-page.txt
sleep 300
goto START
```

Producing data which looks like this:

```
TimeThis :  Command Line :  wget -Q2m --quiet --recursive --convert-links --page-requisites --restrict-file-names=windows --domains widgets.com --no-parent www.widgets.com > junk 2>&1
TimeThis :    Start Time :  Sat Mar 09 08:13:17 2013
TimeThis :      End Time :  Sat Mar 09 08:13:45 2013
TimeThis :  Elapsed Time :  00:00:28.079

TimeThis :  Command Line :  wget -Q2m --quiet --recursive --convert-links --page-requisites --restrict-file-names=windows --domains fhcrc.org --no-parent www.fhcrc.org > junk 2>&1
TimeThis :    Start Time :  Sat Mar 09 08:08:08 2013
TimeThis :      End Time :  Sat Mar 09 08:08:17 2013
TimeThis :  Elapsed Time :  00:00:09.656

TimeThis :  Command Line :  wget -Q2m --quiet --recursive --convert-links --page-requisites --restrict-file-names=windows --domains fhcrc.org --no-parent www.fhcrc.org > junk 2>&1
TimeThis :    Start Time :  Sat Mar 09 08:13:17 2013
TimeThis :      End Time :  Sat Mar 09 08:13:45 2013
TimeThis :  Elapsed Time :  00:00:28.079

…
```

Case 1: Examine data

Crunch Numbers

	10Mb/s	**100Mb/s**
Sample Count	2693	197
Average	12.8	15.9
Median	11.4	10.5
Max	39.4	144
Min	9.4	2.4

Would upgrading to a 100Mb/s TLS improve the responsiveness of Web browsing?

Case 1: Your Story

How would you tell this story?

What clues do trace analysis and simulation offer?

How are these approaches flawed? What do these approaches fail to consider?

Will upgrading the TLS to 100Mb make a difference, and if so, by how much?

Case 1: My Story

Network Impact and Application Assessment jobs are hard; trace analysis and simulation offer only hints.

Still, I use these techniques to offer clues as to where to look for the major contributors to a performance problem.

Case 2
Intermittent File Copy Failures

The Network Sucks

Case 2: Background

The MIS group dumps their SQL Server databases to files every few hours and copies them into a ring buffer on the Solaris box *sam*: this is a snapshot strategy, allowing them to rapidly recover Prod, Dev, or Test to a previous state with some granularity. They have hundreds of similar processes running throughout the day/week/month across their many dozens of hosts.

The MIS group detests *sam*: for years, intermittently, these processes fail with *A network error has occurred* or *The server is no longer available*. Their Tidal job scheduler pages them when these fail (7x24), they have to restart the jobs manually … in testing, they try copying database dumps and other files to one of their own boxes, and it works reliably … but they don't have enough space on their own gear … their SAN is mostly full … their requests for more space have been denied for years … they have to beg for scraps on *sam*, which front-ends gobs of disk used for scientific functions … *sam* runs yucky Unix, whereas sensible sys admins run Windows … it's just a mess. To top it all off, this problem has been getting worse the last few months.

Management comes to you and says, look, these folks are whiners; they are forever telling us the network is bad, DNS is bad, AD is bad, *sam* is bad, the sky is falling, they need a second SAN … make them go away.

J4 Data Center
120 KW Facility
~600 MAC addresses
~1200 IP addresses

j4sr-x-esx are Layer 2 only
All ports belong to one VLAN (42)

ja-x-rtr are Layer 3 only
Four /23 subnets are layered onto VLAN42

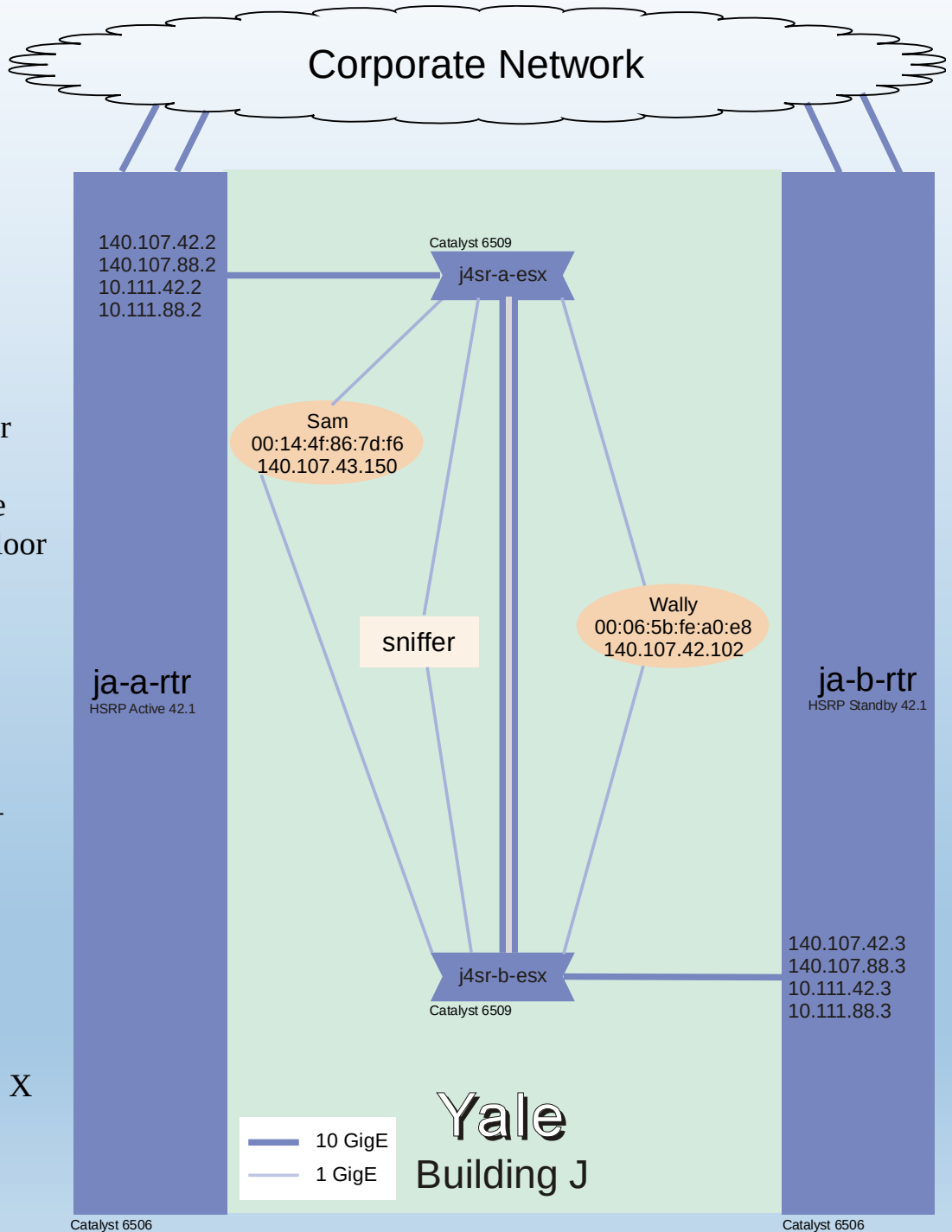
ja-x-rtr feed a handful of additional access-layer switches (not shown), one per floor in the J Building. One VLAN <=> one Subnet <=> one Switch: VLANs do not metastasize past their floor

The Corporate Network contains ~six other buildings of similar design

Sam services nearly a thousand clients via NFS and SMB. This Data Center hosts numerous services, including home directories for ~2500+ users, e-mail company-wide, a VMWare farm, many others

The MIS folks are the only ones reporting problems

Popular OSes include Windows, Linux, and OS X



Case 2: Problem Statement

Initial Problem Statement

> Our jobs fail because the network is flakey, and *sam* is heap of junk

Improved Problem Statement

> File copies from any of several dozen servers to *sam* intermittently fail with the error message *A network error has occurred* or *The server is unavailable*

Can you think of a better Problem Statement?

Case 2: Whine

The Sniffer is overwhelmed on the *Sam* capture -- even with slicing & filtering, it drops lots of frames and quickly exhausts disk space

The problem is intermittent and happens to other MIS boxes … but not to the one we pick … we finally shift to *Wally*, which seems to experience the problem more frequently …

We cancel the *Sam* capture … but the *Wally* capture fills up disk rapidly, too … by the time the alarm reaches us, the relevant pcap has been overwritten

We finally configure Swatch (Perl script which watches *syslog* on *loghost*) to watch for the Tidal 'job failed' message, page us, and then we move fast to stop the sniffer. For grins, we configure *Wally* to ping *Sam* continuously, using a dedicated Cmd window

This all takes weeks

Finally, the Tidal job failure message hits your pager on 2009-07-22 around ~14:50, you jump, and you stop the sniffer … and yes! the pcap for this window is still there

Case 2: How Would You Tell the Story?

Who will recap for us?

Case 2: Problems

Unable to capture Definitive Diagnostic Data

This is a Rapid Problem Resolution term: we did not have the tools to instrument the path from client to server: had we captured simultaneously at *Sam* and *Wally*, the problem would have yielded more rapidly.

Propose

Add this to the list of risks tracked by Problem Management

Insufficient tools and access to diagnose client/server problems in the J4 data center

Case 3
Router Misses Pings

Moles Rule

Case 3: Background

The IT team uses a range of applications running on their management hosts *jargos* and *jariel* to monitor their environment.

Two mgmt applications, *NodeWatch* and *watch-via-arp*, intermittently report that the default router has failed to return a ping, leading to alarms, which quickly clear but which trouble the on-call staff (particularly troubling in the middle of the night!) The dev behind these applications has modified them to hold the alarm for a few seconds and then try the ping again (success invariably follows); this hack suppresses the alarms but merely masks the problem. The secondary management stations, located in another data center, also ping the relevant router interfaces and never report missing a ping.

Do we really care that a router is missing the occasional ping?

On one level, not really – user applications don't depend on pings. On the other hand, the brains inside a router (the Cisco kids call this a Route Processor in their Catalyst 6500 platform) do a lot more than respond to pings; they also respond to HSRP / EIGRP / UDLD Hellos, ARP Requests, and other fundamental functions underlying transport services. If one of these other functions is getting dropped, then we could encounter a range of problems.

And we're talking about our largest data center here.

Case 3: Background

We've been cavalier about VLANing in this data center – everyone lives on the same VLAN. Sure, we use secondary subnetting to split up the IP space .. but the broadcast domain encompasses everyone.

If pings aren't getting through … what about ARPs? And what happens to all those dual NIC servers, with their high-availability schemes which rely on broadcasting or multicasting Hellos to each other … or ARPing for the default router? What if those start getting dropped, and those servers start thinking that one NIC or the other has gone bad?

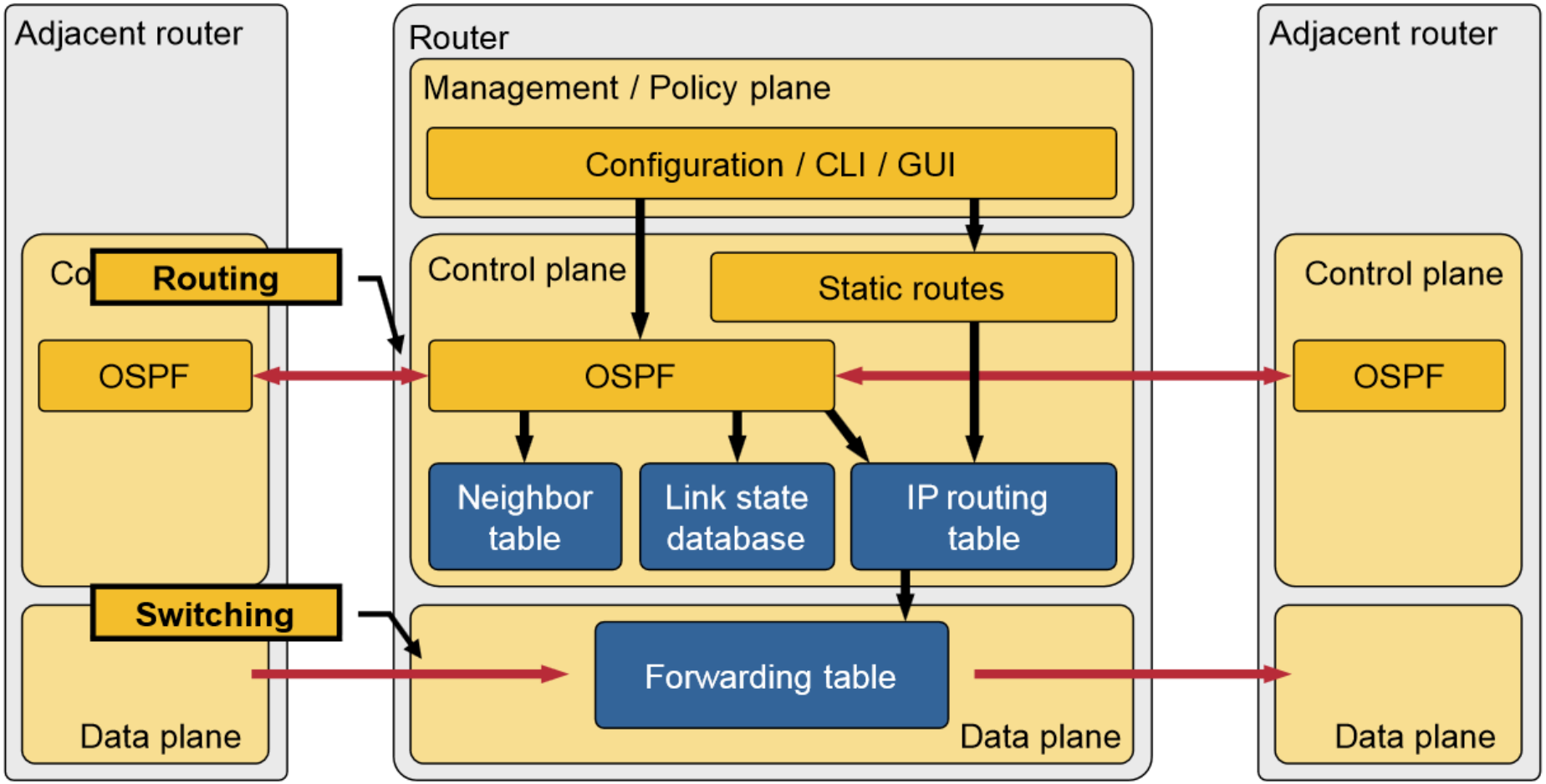
If HSRP Hellos don't get through, then the Active/Standby status of the redundant routers could start toggling …

But let's back up: maybe the network is discarding all sorts of traffic – Data Plane as well as Control Plane – and we're just noticing the Control Plane frames. Perhaps *j4sr-x-esx* are overwhelmed and are tossing all sorts of stuff …

There's plenty of room here for the neurotic tech to worry …

Lingo

Management, Control and Data Planes



Graphic cribbed from Ivan Pepelnjak: http://www.ipspace.net

Case 3: Reading Log Files

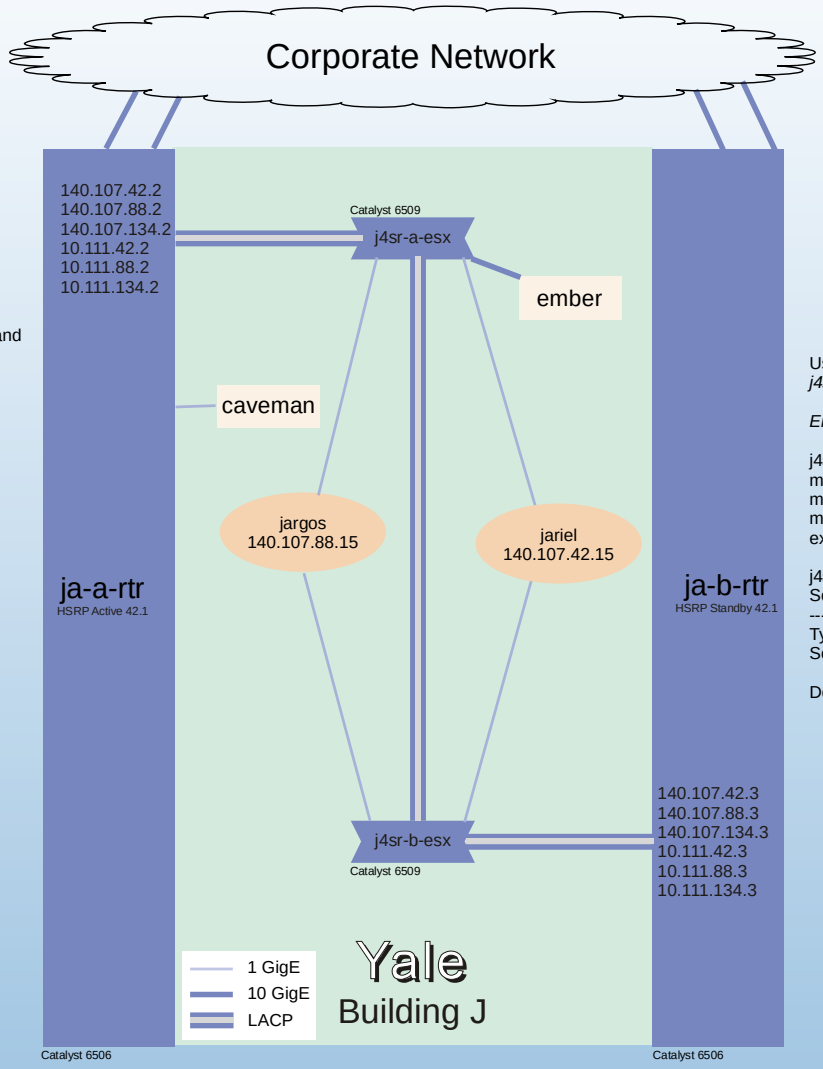
See nodewatch-and-watchviaarp.txt

Here is *watch-via-arp* reporting that the default router did not return a ping:

```
Jan  1 01:35:29 jargos root /opt/vdops/script/watch-via-arp[2998]: ja-x-rtr-v42
/ 140.107.42.1 did not return a ping
```

Here is *NodeWatch* reporting that the default router did not return a ping:

```
Jan  3 14:59:36 jariel nodewatch: Essential nodes missed a ping:  ja-x-rtr-v42
```



Use *Caveman* to capture on both the Route and the Service Processors

Caveman is a Shuttle PC equipped with an Intel Pro/1000 SX card

```
ja-a-rtr# config t
monitor session 1 type local
  description Capture RP to Caveman
  source cpu rp
  destination interface Gi6/2
exit

ja-a-rtr# sh monitor session 1
Session 1
---------
Type                : Local Session
Status              : Admin Enabled
Description         : Capture RP to Caveman
Source Ports        :
    Both            : rp,sp
Destination Ports   : Gi6/2

Egress SPAN Replication State:
Operational mode    : Centralized
Configured mode     : Centralized (default)
```

Use *Ember* to capture on the Portchanneled uplink between *j4sr-a-esx* and *ja-a-rtr*

Ember is a Fluke Optiview XG listening to a SPAN port

```
j4sr-a-esx# config t
monitor session 1 description SPAN uplink to Ember
monitor session 1 destination interface Te6/2
monitor session 1 source interface PortChannel1
exit

j4sr-a-esx# show monitor session 1
Session 1
---------
Type                : Local Session
Source Ports
    Both            :  Portchannel1
Destination Ports : Te6/2
```

J4 Data Center

120 KW
~600 active MAC addresses
~1200 active IP addressees

HSRP

ja-x-rtr are HSRP Partners
  Hello 3, Timeout 1
Generally, *ja-a-rtr* is HSRP Active and thus
  owns 140.107.42.1, 140.107.88.1, 140.107.134.1 ...

VLANs

All Ports on *j4sr-x-esx* belong to the same VLAN: VLAN42
The VLAN42 interfaces on *ja-x-rtr* support secondary IP addressing
  140.107.42.0/23
  140.107.88.0/23
  140.107.134.0/23
  10.111.42.0/23
  10.111.88.0/23
  10.111.134.0/23

Case 3: Problem Statement

Initial Problem Statement

The network throws away packets

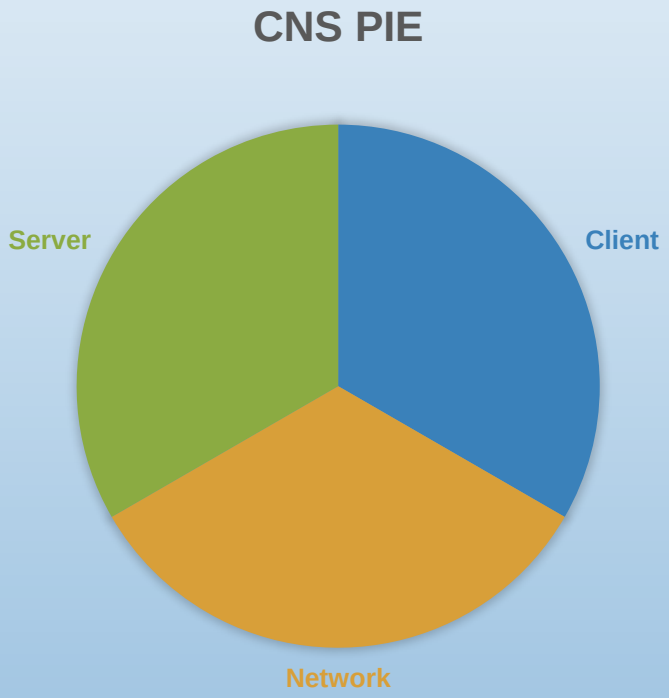
Improved Problem Statement

ja-a-rtr-v42 intermittently misses pings … and we're worried

Can you think of a better Problem Statement?

Case 3: Narrow the Fault Domain

How to narrow the fault domain? Recall the Client – Network – Server Pie … who is dropping the frames?



- Are the Clients *jariel* and *jargos* dropping the outbound pings?
- Is the Network dropping the pings?
- Is the Server *ja-a-rtr* dropping the pings?

Case 3: Traces

By deploying sniffers in different locations, we develop insight into whether or not the ICMP Echos are reaching *ja-a-rtr*

See the Events folder. Each directory encapsulates traces captured on that day.

When you look at a trace full of pings, how do you find the ICMP Echos for which there is no corresponding ICMP Echo Reply?

```
not (icmp.resp_in or icmp.resp_to)
```

Case 3: Your Story

How would you tell this story?

Case 4
The Internet is Slow

Must be the Firewall

Case 5
The Web Server is Slow

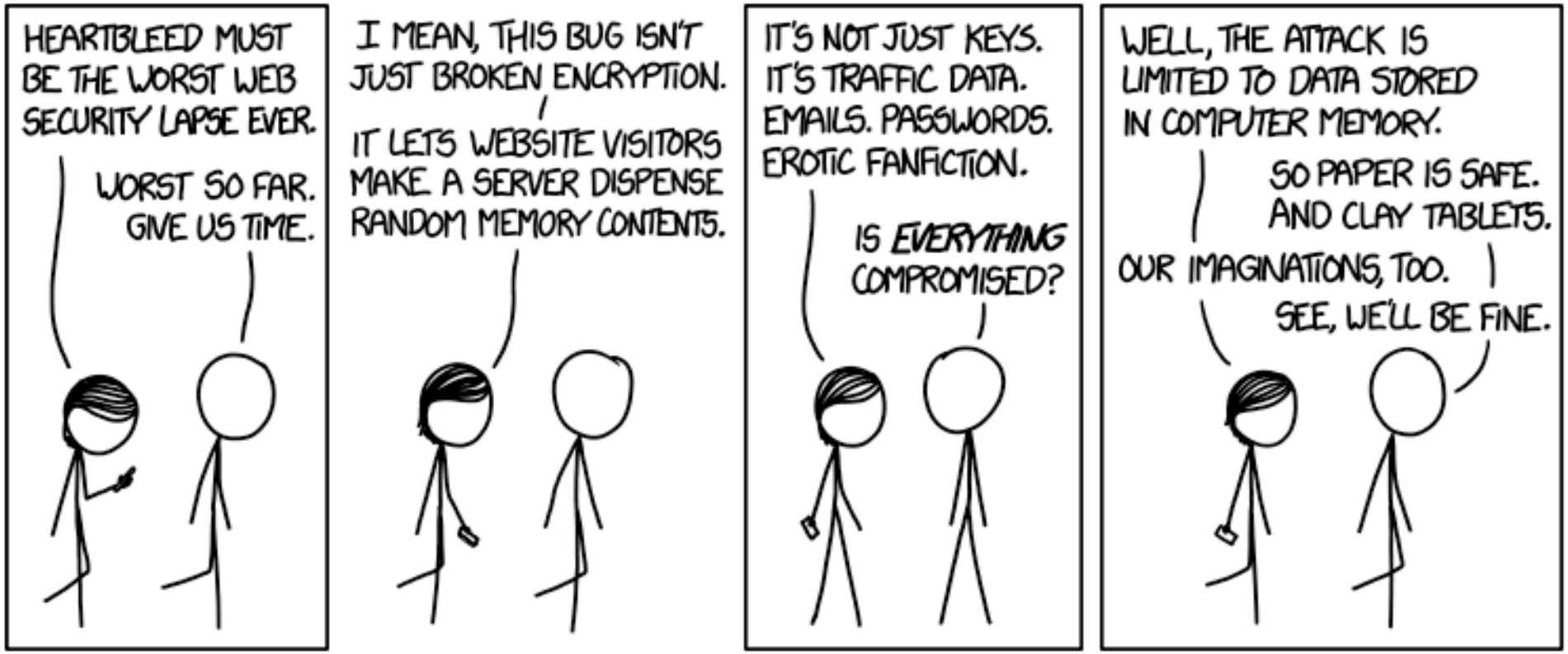
Mouse Brains

Case 6
Intermittent Citrix Disconnects

Xenophobia

HEARTBLEED

|< < PREV RANDOM NEXT > >|



Wrap-Up

Questions, Comments, Complaints?

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