Upgrading Our Network Analysis Tools

OVERVIEW

This document describes possible upgrade paths for our sniffing capabilities.

Packet capture is one of many tools we employ when performing 'network analysis', the art & science of analyzing why clients and servers are having trouble speaking with one another. I claim that we solve the vast majority of client/server issues using tools other than packet capture. In fact, most techies either don't know how to capture packets or don't feel comfortable doing this … but they manage to resolve stacks of client/server issues just fine without this capability. Even the experienced network analyst tends to resolve problems with minimal sniffing … because sniffing tends to be time-consuming and resource-intensive. In our environment, we tend to deploy packet capture on problems which have proved intractable via the usual methods.

Currently, we own three dedicated devices for performing packet capture. These devices have various capabilities and limitations. Two are Shuttle PCs equipped with extra NICs; the third is a laptop bundled with specialized packet capture hardware. All three are equipped with specialized software.

I claim that our current tool kit allows us to tackle most problems. So why consider buying anything new? The primary benefit to augmenting this kit with additional tools would be reducing Mean Time To Repair (MTTR). Figuring out whether the cost of these tools is worth the reduced MTTR is the decision which this document tries to inform.

SNIFFING PRIMER

There are many ways to capture packets; each approach carries its own set of pros and cons. If you want a technical discussion of these approaches, read on. If you want to fast-forward across the geeky details, skip this section.

Typically, network analysts start by narrowing down the fault domain: should we focus our attention on the Client, the Server, or the Network? As part of this process, the analyst often wants to perceive the experience from the client's point of view, from the server's point of view, and from the network's point of view and then compare all three points of view, to see where the potholes may lie. Acquiring such a perception can involve deploying packet capture gear at the client, at the server, and inside the network.[1]

In our environment, we generally skip sniffing from the Network's point of view, because we have designed our network to deliver packet transport services only; except during a few worm outbreaks, we leave the other fancy packet-messing features disabled[2]. This means that our switches and routers forward what they receive; they don't filter packets in transit nor do they

[1] And, of course, it often involves gathering data via other means at the client, the server, and the network: the output from various "show" commands on switches & routers, the output from various 'grep' commands on syslog, the output of 'netstat', etc. on clients and servers, and a myriad other sources.

[2] We believe this is a Good Thing. However, not all network engineers agree with us; some networks are designed to do some serious packet messing. In such a network, I would want the ability to sniff inside the network, i.e. between switches & routers.

modify them (other than the usual MAC address swapping which routers perform, by definition). As a result, most of the problems 'inside' our network tend to be tractable using the tools on-board our gear.

The exception to this is at the borders of the network, where we install Intrusion Prevention Devices and Firewalls. These packet transport devices do some serious messing with packets, and we regularly find a need to sniff on either side of them -- i.e. in the middle of the network -- in order to see what they are doing.

Terminology

COPPER

Any network transmission media comprised mostly or entirely of Cu atoms … the Category 5 or 6 patch cords we use … the Cat 5 or 6 wiring in the walls. Typically, we use copper to plug end-stations in to the network, i.e. to deliver the 'access layer'.

GLASS

Any network transmission media comprised mostly or entirely of Si atoms mixed with impurities to create glass. [Typically called 'fiber optic cable' or 'fiber' for short … I'm trying to avoid the use of the word 'fiber' right now, as I'm sensitive to how heavily overloaded this term has become in our environment.] Typically, we use glass to tie network devices together, i.e. to interconnect switches with routers. Technically, glass provides more 'bandwidth' than does copper … however, at the Hutch, we don't utilize this property.[3] Instead, we deploy glass when we need distance … Ethernet signals tend to become unreadable after about 100 meters on copper, whereas on glass, Ethernet signals can travel for multiple kilometers (and even tens of kilometers, given the appropriate optics).

PROBE

VDOPS invented term for a PC built from off-the-shelf hardware dedicated, by policy, to capturing packets. In our environment, synonymous with the two Shuttle PCs (named 'apeman' and 'caveman'), equipped with a 10/100 transport card, a 10/100/1000 capture card, and a dual-ported 1000BaseSX capture card, loaded with Fluke's *Protocol Expert* and *Ethereal*.

[3] *Bandwidth* refers to the maximum theoretical amount of information per second which the media can convey; *throughput* refers to the actual number of bytes per second being carried. Gigabit Ethernet delivers 1Gb/s *throughput*, regardless of whether it is carried over copper or glass (or barbed wire or air or carrier pigeon … although the carrier pigeon's feathers tend to become a bit mussed, when handling GigE traffic!) The *bandwidth* of glass greatly exceeds that of a copper … but since, in our environment, we are only loading a single Gigabit Ethernet stream on each, the resulting *throughput* is identical. *Bandwidth* tends to become relevant in service provider networks, where the carrier will load multiple 1Gb/s streams of traffic on a single glass cable, transmitting each one at different wave lengths and thus utilizing glass' superior *bandwidth* to deliver more bytes per second of *throughput*.

POD

VDOPS invented term for specialized packet capture hardware which permits (a) full-duplex sniffing, (b) line-rate capture, and (c) accurate time-stamping. Three manufacturers (Xyratex, Endace, and Finisar) manufactur the specialized Ethernet NICs used in Pods -- a range of vendors bundle these NICs into luggable PCs or rack-mounted boxes or simply re-sell them directly … along with the drivers and software needed to use them. These NICs typically contain a hundred or two hundred megabytes of on-board RAM, clocks accurate to the nanosecond, and specialized hardware for performing write-to-memory or write-to-disk functions without needing to invoke the host PC's CPU.

LINE-RATE

The ability to capture packets as fast as the wire sees them … e.g. a line-rate gigabit Ethernet sniffer can capture a two gigabits per second (one gigabit in one direction plus one gigabit in the other direction). Most sniffing solutions drop packets when traffic levels surpass a certain rate, where that rate is a function of a myriad interactions between NIC performance, NIC driver performance, operating system design, bus performance, and CPU resources.

PACKET SLICING

If you're awake, you're probably wondering where a Line-Rate sniffer puts two gigabits per second of packets … that fills up main memory in your average PC pretty darn fast. And it fills up disk fast, too, assuming that the sniffer is capable of writing to disk. To solve this problem, sniffing software allows one to capture only the first X bytes of a packet, typically 64, 128, 384, or 512 bytes, in order to reduce RAM/disk utilization. Naturally, one loses visibility when performing packet slicing … if the part of the packet containing the clue to the problem lies in the 'sliced' part, then the analyst can't see it.[4]

Myself, I don't do this voluntarily … I don't like missing data. However, this trick becomes useful when pairing a low-end Probe with a fast stream of data: the low-end Probe can't pull a big stream of data off the card fast enough, so it drops packets. With packet-slicing, it has less work to do and can capture packets at a faster rate.

Even with Pods, this trick becomes useful when analyzing fat streams of data, because it reduces the size of the resulting trace, making it easier to manipulate. (When I'm analyzing backup traffic … I don't usually need to see the contents of the packets … I just need their headers. One second of gigabit-rate backup traffic burns 100MB of disk space … but only 4MB with aggressive packet slicing turned on.)

Regrettably, when analyzing SMB traffic (Microsoft's protocol), packet slicing is less useful … SMB headers are so huge, that one has to slice at 512 bytes in order to get everything (whereas slicing at 64 bytes or sometimes 128 bytes is enough when analyzing other protocols).

[4] For example, one loses visibility into a popular problem in which an end-station, due to NIC failure or a bug in NIC driver software, starts mis-calculating the TCP checksum … without the full TCP frame, the sniffer can't verify the TCP checksum and thus becomes blind to this issue.

SPAN

Switched Port ANalyzer. This is a function of a switch (Ethernet and Fibre Channel), which instructs the switch to xerox packets traversing a specified port and to send copies to another port where, presumably, the analyst has plugged a Probe or Pod. SPAN ports do not see precisely the same traffic stream that the original port sees[5], and sometimes this is a problem.

TAPS

In-Line Taps

In-Line Taps are devices which pass packets through themselves, "xeroxing" them as they go by and forwarding the copies to the analyzer. In-line Taps have two ports pointing toward the analyzer: one carrying Transmit traffic, the other carrying Receive traffic. Only Pods contain the specialized hardware needed to take advantage of an In-Line Tap (because only specialized Ethernet NICs can receive traffic on *both* the receive *and* the transmit pairs of wires).

Aggregation Taps

The Aggregation Tap passes packets through itself but instead of spitting Tx packets out one port (toward the analyzer) and Rx packets out the other (toward the analyzer), it combines the two traffic streams and delivers a single procession of packets out a single port. This allows the analyst to plug off-the-shelf hardware, a Probe, into the Aggregation Tap. Of course, if the combined procession of packets exceeds media speed (say, if the wire is carrying Fast Ethernet traffic at the rate of 50Mb/s in one direction and 51Mb/s in the other), then the resulting procession runs the risk of overflowing the buffers on board the Aggregation Tap … at which point the Aggregation Tap will start tossing packets, before the analyzer has a chance to see them.

In a further wrinkle, some Aggregation Taps support EtherChannel.

General Statements about Taps

Glass Taps tend to employ mirrors or crystals, splitting off some of the light and forwarding it to the analyzer. This decreases the distance which the production light can travel. Glass Taps tend to be passive, i.e. to require no power.

Copper Taps tend to employ powered electronics which they use to xerox the packets. When they lose power, they continue to forward production traffic but lose the ability to xerox packets.

CAPTURE FILTERS

When one knows enough about a problem to start narrowing the fault domain (i.e. you're only concerned about traffic passing to and from a single IP address, for example), one configures a 'Capture Filter' -- this tells the sniffing software to grab only packets meeting the criteria

[5] Damaged packets (packets whose Ethernet CRC check fails), for example. Or packets which the switch is administratively configured to drop.

specified in the filter and to discard the rest. This reduces the size of the resulting trace which in turn makes it easier to manipulate.

DISPLAY FILTERS

Once one is analyzing a trace, one often configures a 'Display Filter' to either hide packets which one has determined play no role in the problem or to highlight packets which one wants to analyze further.

As Barry Banner (colleague of Mike Pennachi) says "I can solve most problems by analyzing just a couple packets … perhaps a dozen at most … but extracting those dozen packets from the tens of thousands I've captured, that's the hard part." Filters are the key to extracting these dozen packets.

SPECIALIZED NIC DRIVERS

The average NIC driver discards packets containing Ethernet-layer errors. From a network analyst's point of view, this is undesirable: the network analyst *wants* to see damaged packets.

Some vendors produce their own NIC drivers which overcome this flaw, capturing even damaged packets and hogging machine resources in order to reduce the chances of dropping a packet. All Pods ship with such NIC drivers; as far as I know, only Network General ships such drivers with their software-only solution (and for only a small selection of NICs at that).

VISUALIZATION SOFTWARE

High-end vendors, notably Network General[6], produce software which analyzes traces (sometimes lots of traces) and offers tools which provide meta-analysis -- insights into what might be happening *before* you start diving into the packet details. I don't know much about this software -- it tends to be expensive … although Ethereal includes some visualization tools. These tools help the analyst to pick out patterns faster than a packet-by-packet walk allows.

EXTRACTION SOFTWARE

Some analysis software delivers tools for extracting packets from a large trace … for specifying, for example, "I want to see all packets to or from IP address a.b.c.d from 9pm last night to 1am this morning". This capability is essential for manipulating traces files which are larger than the physical memory of your workstation and useful when analyzing large traces in general (analyzing large traces is CPU-intensive).

CAPTURE-TO-DISK

This feature of packet capture software allows saving packets to disk when the memory buffer fills. This allows one to capture more traffic than physical memory can hold. Pods generally

[6] And many others, who focus on analysis rather than capture, including NetIQ, Optimal Networks, NetMetrix …

ship with the ability to capture-to-disk *without* interrupting capture, whereas Probes generally stop capture, save to disk, and then resume capture.

TIME-STAMPING

All capture software records the time when the packet was captured … but the clock in an average PC is accurate to 10^{-3} seconds, whereas packets arriving at gigabit rates can arrive every 10^{-7} seconds. Specialized packet capture hardware, typically built into Pods, employ clocks accurate enough to deliver this kind of time-stamping. Accurate time-stamps are essential for analyzing some client/server issues, useful for analyzing others, and irrelevant for analyzing most issues.

Packet Capture

INSERTION METHODS

In all sniffing situations, one needs a place to put the packet capture software where it will see (and capture) the packets being exchanged by the client and the server. In other words, one needs an *insertion point*. Figuring out where and how to insert involves expertise and, often, specialized gear.

Here are the various approaches, along with their capabilities and drawbacks. I use *Ethereal* as my example software package, since it is free … but any of a selection of commercial packages could of course be used instead.

Client Based

Install Ethereal on the client.

Server Based

Install Ethereal on the server.

Probe + Hub at Client or Server

A probe is typically a portable PC (Shuttle PC or laptop) loaded with sniffing software. The Hub is typically a small 10/100 (switching) Ethernet hub. One plugs the Client (or the Server) into the Hub, along with the Probe, and then attaches the uplink port on the Hub to the rest of the network. For the purposes of this document, Probe's are half-duplex devices: i.e. they are equipped with off-the-shelf NICs which capture packets arriving on their Receive wires while ignoring their Transmit wires.

Pod + In-Line Tap at Client or Server

For the purposes of this document, Pods consist of specialized hardware: NICs which are designed to capture packets arriving on *both* their Receive wires *and* their Transmit wires.[7] Furthermore, this specialized hardware is engineered so that it can capture packets at "line-rate",

[7] As far as I can tell, there are three manufacturers of these NICs: Xyratex, Endace, and Finisar.

without fear of dropping packets due to resource constraints.[8] Finally, this specialized hardware is equipped with precise clocks, which permit the accurate recording of packet arrival times. In general, Pods are accompanied by In-Line Taps. Pods, by definition, are equipped to receive these two streams of traffic and combine them intelligently.[9]

Probe + Aggregation Tap at Server

This solution bridges the price distance between Probe + SPAN/Hub and Pod + In-Line Tap. It allows the analyst to precisely view the Client/Server conversation while deploying an inexpensive Probe (rather than a high-end Pod). However, at high traffic rates, the Aggregation Tap will lose packets.

Always On

These devices tend to be Pods bundled with lots of disk (hundreds of gigabytes or even multiple terabytes). This allows the device to save what it sees for minutes, hours, days, or even weeks, depending on how much traffic passes by. In this way, the analyst can "go back in time". Higher-end boxes contain multiple ports plus the smarts to use them, allowing them, for example, to support EtherChannelled server connections (multiple paths) or multi-pathed networks (like our dual uplink design off server rooms).

Big Boxes

I don't have names for these devices … but they are characterized by big price tags and non-trivial weight. One tends to install them permanently and then route high-traffic links through them. They can perform line-rate capture and can "glue together" conversation flows from multiple paths. For our purposes, they look like multi-port Pods.

Permanently Installed Taps

Analysts like Taps because, once installed, they permit packet capture without messing with the Client, the Server, or the Network (SPAN ports on the Ethernet switch). Walk up to the Tap, attach the Probe or Pod, and start sniffing: no downtime, no fuss, no muss. However, inserting the Tap is a pain -- runs the risk of disrupting users. The Permanently Installed Tap fixes this problem. We have done this around both the FHCRC and the SCCA firewalls -- these permanently installed In-Line Taps allow us to insert a Pod without risk of disrupting production traffic. Permanently Installed Taps are just portable Taps with a rack-mount kit.[10]

PROS AND CONS

Name	**Pros**	**Cons**
Client Based	▪ Cheap	▪ Requires modifying the Client

[8] In general, the manufacturer accomplishes this by installing dedicated RAM on the NIC and by providing a hardware-assisted function for copying those packets out of this dedicated RAM and into either main memory or onto disk, at gigabit rates.

[9] Some pods have In-Line Tap functionality built into them.

[10] At the high-end, where density is paramount, manufacturers produce devices which look like Ethernet switches but which are really high-density taps. These devices tend to live outside our price range, so I ignore them here.

	▪ Easy	▪ Cannot capture boot traffic ▪ Sometimes misses packets due to client resource limitations ▪ Misses physical layer errors and NIC-induced errors like TCP checksum errors ▪ Requires physically visiting the Client[11]
Server Based	▪ Cheap ▪ Easy ▪ Easily remote controlled	▪ Requires modifying the Server ▪ Cannot capture boot traffic ▪ Sometimes misses packets due to server resource limitations. This effect becomes more prominent as traffic rates and CPU utilization increase ▪ Competes with production services for resources -- this can change the character of the problem ▪ Misses physical layer errors and NIC-induced errors like TCP checksum errors
Probe + Hub at Client	▪ Allows the analyst to watch the end-user's experience	▪ Requires gear: probe + hub ▪ Requires physically visiting the Client ▪ Requires finding space & power at the Client location ▪ Changes the connection from full-duplex to half-duplex; this can change the character of the problem ▪ Probe drops packets at high traffic levels ▪ Only works with 10/100BaseTX[12]
Probe + Hub at Server		▪ Requires gear: probe + hub ▪ Requires physically visiting the Server ▪ Requires finding space & power at the Server location ▪ Changes the connection from full-duplex to half-duplex; this

[11] Under some circumstances, one can remotely control the Client …. but generally … the interferes with the end-user experience.

[12] While the IEEE 802.3 specification defines Gigabit Ethernet hubs … I know of no vendor that has ever implemented them in product.

		can change the character of the problem ▪ Requires unplugging a Server NIC and plugging it back into the hub ▪ Requires redirecting all Server traffic across the single NIC plugged into the hub ▪ Probe drops packets at high traffic levels ▪ Risks disrupting users ▪ Only works with 10/100BaseTX
Probe + SPAN: Client	▪ Shields the end-user from the extra gear	▪ Requires administrative access to the Ethernet switch ▪ Supports sniffing on multiple hosts simultaneously ▪ At high traffic rates, runs the risk of losing packets on both sniffed traffic and other production traffic due to switch buffer overflows ▪ If the sum of both transmit and receive exceed link speed, then the SPAN port will drop packets ▪ Misses physical layer errors ▪ Incorrectly includes packets which the switch administratively drops before transmitting to the client ▪ Messes with time-stamps ▪ Most switches permit only a single SPAN session active at a time
Probe + SPAN: Server		▪ Requires administrative access to the Ethernet switch ▪ Supports sniffing on multiple hosts simultaneously ▪ Supports EtherChannelled links ▪ At high traffic rates, runs the risk of losing packets on both sniffed traffic and other production traffic due to switch buffer overflows ▪ If the sum of both transmit and receive exceed link speed, then the SPAN port will drop packets

		▪ Misses physical layer errors ▪ Incorrectly includes packets which the switch administratively drops before transmitting to the client ▪ Messes with time-stamps ▪ Most switches permit only a single SPAN session active at a time
Pod + In-Line Tap: Client	▪ Leaves the Client entirely untouched ▪ Most closely replicates the Client view point (full-duplex, accurate time-stamps, unlikely to miss packets) ▪ Line-rate	▪ Expensive ▪ Requires visiting the Client ▪ Requires finding space & power at the Client location
Pod + In-Line Tap: Server	▪ Leaves the Server entirely untouched ▪ Most closely replicates the Server view point (full-duplex, accurate time-stamps, unlikely to miss packets) ▪ Line-rate	▪ Expensive ▪ Requires visiting the Server ▪ Requires finding space & power at the Server location ▪ Requires unplugging a Server NIC and plugging it back into the Tap ▪ Requires redirecting all Server traffic across the single NIC plugged into the Tap
Probe + Aggregation Tap: Server	▪ Leaves the Server entirely untouched ▪ Cheaper than the Pod + In-Line Tap approach ▪ Some models support EtherChannel (Indigo)	▪ Requires visiting the Server ▪ Requires finding space & power at the Server location ▪ Requires unplugging a Server NIC and plugging it back into the Tap ▪ Requires redirecting all Server traffic across the single NIC plugged into the Tap ▪ Tap drops packets at high traffic levels

Packet Analysis

Once you've captured the packets, you want to look at them, to help refine your focus (client, network, or server) and to better understand what might be happening. Doing this requires firing

up packet analysis software. Most packet capture software performs double duty as packet analysis software.

Analysis Software tends to be quirky -- the interface to each package is wildly different from the next, and humans tend to develop preferences. In addition, each analyzer contains capabilities unique to that product, useful tools for understanding what the problem might be which are available *only* in that analyzer.

Here are some generalizations about capabilities:

ETHEREAL (OPEN SOURCE)

Some of the best decodes around, the subject of rapid development and frequent updates; excellent decodes, FREE. However, Ethereal is unstable -- its GUI will become confused (requiring a quit and a reload), and it will crash while capturing and while analyzing. Once a trace file exceeds ~10MB, Ethereal starts slowing down, and after ~30MB, I tend to switch to another analyzer, because the time involved in manipulating the file exceeds my patience time and because Ethereal's penchant for scrambling its GUI or crashing becomes too frequent for my taste. [However, I grit my teeth and bear it when I want to analyze the trace using one or more of Ethereal's visualization tools -- Ethereal is the only package we own which includes such tools.]

Permits command-line filtering (a boon for fast typists), the ability to read and write many analyzer formats, and a range of 'visualization' functions.

SNIFFER (NETWORK GENERAL)

Excellent decodes, often augmenting or occasionally even exceeding Ethereal's. Fast -- performance scales in a less-than-linear way with trace file size!

SURVEYOR (FINISAR)

Inexpensive, range of advanced functions, makes it easy to control multiple analyzers from a single copy of the software. Lousy decodes … however, the VoIP module is an exception to this. Fluke has OEMed Surveyor; their version is called Protocol Expert -- the two are interoperable. The next version (version 7.5) promises to allow the operator to substitute Ethereal or Sniffer's decodes for Finisar's.

OMNIPEEK (WILDPACKETS)

We don't own a copy of OmniPeek -- but I saw a demo recently and thought its interface beat anything else I've seen hands-down. It has the same ability Surveyor has, to control multiple analyzers from a single copy of the software. And, it allows the user to substitute Ethereal's decodes for its own!

Currently, my favorite approach to analyzing a trace is to open the 'good' trace on one monitor, the 'bad' trace on the second monitor, and compare … using Ethereal … because I prefer its interface. If I want more input on packet decodes, I switch to Sniffer … and if the traces files are large, I try to stick with Sniffer. Sean uses Ethereal and Protocol Expert. Jonathan uses Sniffer primarily.

WHAT WE OWN TODAY

We own two Shuttle PCs equipped with multiple NICs (one 10/100 for remote control, one 10/100/1000TX for capture, two 1000BaseSX for capture). When these boxes boot, they register with WINS, and we then employ the built-in Windows Remote Desktop applet to control the device remotely, launching either Protocol Expert (commercial packet capture software from Fluke) or Ethereal (open-source packet capture software) to capture packets. Post-capture, we copy those files off to our workstations, where we analyze them.

We own a Finisar THG Notebook System, an external PCI bus containing a pair of Finisar's specialized Ethernet NICs glued to a laptop running Surveyor (Finisar's commercial packet capture application) which permits line rate/in-line sniffing. Stuffed into the 'accessories' bag accompanying this laptop is one 10/100BaseTX In-Line Tap, one 1000BaseTX In-Line Tap, and one Glass In-Line Tap.

We own two copies of the Sniffer software (Jonathan and Stuart). We also own a Sniffer WAN pod (a specialized piece of hardware which permits us to sniff on serial links, like T1s).

USE CASES

Client-Side Sniffing

Generally, once an issue escalates to the packet capture phase, we take the next step by acquiring just a Client side packet trace. The user says "The application isn't working" or "The application is slow", and we visit the user to begin characterizing what is happening. Clients are easy to disrupt (you only have to negotiate downtime with one person) and tend to consume little bandwidth, meaning that one can sniff using cheap techniques: Ethereal installed on the Client or a low-end Probe inserted near the Client (via a hub or via a SPAN port).

Our current tool set is well suited for analyzing issues from the Client side -- we have two Probes; we are thus equipped to analyze two Client-side issues in parallel.

Historical Situations

Frequently, we receive a call saying something like "An hour ago [or last night or yesterday] such-and-such happened -- can you tell me why?" Of course, we cannot deploy packet capture against such a problem -- the packets have already run away!

Our current sniffing tool set cannot tackle historical situations.

Intermittent Situations

Some problems are intermittent. In those cases, we install the Probe and configure it to capture packets from the relevant end-station and then wait for the problem to reoccur.[13] Solving this kind of problem takes persistence on the part both of the analyst and the user.

Our current tool serves us well.

Server-Side Sniffing

In some cases, the Client side analysis leaves room for doubt: are Client-emitted packets reaching the Server? Is the Server emitting a response to the Client? In this situation, we typically want to insert a Probe at the Client side *and* a Probe at the Server side. For many problems, our current tools work fine: we insert one Probe at the Client, we plug the second Probe into the server's Ethernet switch and SPAN the server's port (after asking the Server Admin to route all traffic across one NIC), and then capture two traces simultaneously.

This approach has served us well. The typical server isn't passing so much traffic that enabling the SPAN function overflows switch buffers, nor enough to overwhelm the capture function of the Probes we deploy. Sometimes the traffic levels are a little higher than we like, but judicious use of filtering cuts down the size of the trace to manageable levels.

Our current tool set serves us well.

Zebras

A couple years ago, users were seeing intermittent performance problems with Moe … and the Ethernet port to which Moe was attached was recording an alarmingly high rate of physical layer errors. We could have switched Ethernet ports, cables, and swapped out Moe's NICs … all time-consuming and service disrupting … and had we done this (we didn't), nothing would have changed -- the performance issues would have persisted, and the error counters would have continued to increment.

I deployed a Pod (the Finisar THG Notebook System) and captured sample traffic. This is how we learned of an obscure feature in this line of Intel NIC, a feature which means that the NIC produces damaged packets … but without affecting performance. This is a case where a Pod is the only tool for the job.

Prior to the last few months, this was the only time we have deployed our Pod.

[13] Sometimes, we give the user an icon on his/her desktop; when double-clicked, the Probe starts capturing; when double-clicked again, the Probe quits capturing. Sometimes, we configure the Probe to capture-to-disk, ask the user to record precisely when the problem occurs, and then hope that the sniffer will be capturing at that moment, rather than saving to disk.

Our current tool set is sufficient for tackling such unusual problems.

High-Performance

With the advent of the BlueHeat project, I have been deploying the Pod repeatedly, because I have been suspicious of the Ethernet switch -- I know it is dropping packets -- and because Indigo generates and transmits so much traffic that I have been wary of using our Probes.

I have also started sniffing on backup servers, which receive even more traffic than Indigo does, and this trait also pushes me toward using the Pod, on account of its ability to capture all packets even if the traffic rate should jump to full gigabit.

I have had numerous problems with this Pod. I think I have licked a range of them, having to do with the specialized Windows drivers it employs. However, the product remains flakey (cables come loose and require fiddling plus reboots). Sometimes, the product gets stuck in mode in which it bluescreens whenever one loads the software. Sometimes the laptop just shuts down and can't be restarted without removing the power cord and the battery. And, to date, I have been unable to capture with a filter in place (the product bluescreens when I try). On the other hand, because it is the only tool we have for this job, I have persisted, and I have acquired traces which have contributed to our understanding of various issues.

Our current tool set performs poorly.

Fibre Channel

Fibre Channel was born as a bus for interconnecting peripherals with systems … and grew into a set of networking technologies. In the more familiar networking world, a protocol like HTTP rides inside TCP which rides inside IP which rides inside Ethernet which rides on top of some media, like 1000BaseTX or 1000BaseSX. In the Fibre Channel world, FC4 rides inside FC3 which rides inside FC2 which rides inside FC1 which rides on top of FC0 media. Like all networking protocols, Fibre Channel and Ethernet/IP share techniques like acknowledged vs unacknowledged delivery, flow-control, and error detection. Conceptually they are the same, though of course they are implemented differently.

Thus far, all the packet capture & analysis gear we've discussed works in the Ethernet/IP world. In the Fibre Channel world, the concepts are the same -- but the vendors change … way fewer vendors play in the Fibre Channel space than do in the Ethernet space. To date, I've identified the following as players in this space. Finisar is the acknowledged leader.

Finisar
Xyratex
I-Tech
Ancot
Spirent/Netcom
and possibly Agilent and CATC

For SCSI Host-Based Testing, see the following tools.
I/O Meter (http://www.iometer.org/)
I/O Zone (http://www.iozone.org)
SCSI Tools (http://www.scsitoolbox.com)

Finisar produces a line of gear called the X-gig Analyzer, which is a chassis-based Pod for Fibre Channel. One must also purchase (Fibre Channel-specific) In-Line Taps, of course.

Cisco produces a box which takes SPANned traffic from a Fibre Channel port, rips off the FC1 frame, shoves the rest into an Ethernet frame, and spits it out the box's Gigabit Ethernet port. One would then plug an Ethernet-based Probe into that port. Cisco has also given their Fibre Channel decodes, which they wrote for some other product they sell, to Ethereal … so Ethereal can decode these packets. [Though I know of no one who has written an Ethereal driver for a Fibre Channel card, i.e. I don't know of a way to actually capture Fibre Channel frames using Ethereal.] This strategy allows one to analyze FC2-4 issues but leaves one blind to FC1 issues.

Additionally, Finisar produces network management software, NetWisdom, which functions as a cross between NodeWatch, Nagios, and RRDTool (aka MRTG, aka Cacti), with some predictive capabilities thrown in as well.

To the best of my knowledge, there are no open-source ways to manage Fibre Channel networks.

Salient Issues:
Permanently inserted In-Line TAPs, or the use of the SPAN function, are essential in Fibre Channel networks, where disconnecting disks from their hosts (in order to insert monitoring gear) is not something one does lightly (hosts communicate with their SCSI drives constantly, via a stream of SCSI 'RDY' frames … if these are interrupted for some amount of time, the host assumes that the SCSI device has gone and gets upset).

Our current tool set does not address Fibre Channel.

SYNTHESIS

Ignoring Fibre Channel for the moment, we have the tools we need to analyze virtually every issue which can arise in our environment. What would obtaining additional tools buy us?

Simply, the potential to reduce MTTR. Here's how.

Client-Side Sniffing

Currently, our two probes tend to be fairly well utilized -- one tends to sit in cf-114, the other tends to sit in j4-401. Thus, responding to a new analysis request requires traveling to the location of one of these probes, packing it up, moving it to the user's location, and proceeding from there. In addition, sometimes when we are analyzing an issue, we want to capture simultaneously, at the Client location and at the Server location -- when we do this, we tie up both Probes.

If we added a third Probe, we could increase the chances that a Probe would be sitting on the shelf, ready to be re-located to the user in question.

Server-Side Sniffing

If the Server, or Firewall or other Device, we are analyzing happens to reside in cf-114 or j4-401, we can start sniffing almost immediately -- we remotely jack into the Probe located there, configure a Capture Filter, log into the relevant Ethernet switch and configure a SPAN session, and start capturing packets.

However, if the Server or other Device isn't located in one of those rooms … or if the usually local Probe has been redirected to an end-user location, then we have to find an unused Probe, pack it up, deliver it to the new location, and proceed from there.

If we dedicated Probes to our largest equipment rooms, we could reduce this shuffling effect .

High-Performance Sniffing

If the Server or other Device pushes lots of packets … where "lots" is *not* a well-defined quantity … then we deploy the Pod. This is a bear -- installing it takes non-trivial effort, on account of the various parts (the external pod with hi-speed serial cable to the laptop and separate power cord, the In-Line Tap plus its power cord, the *four* Cat5 cables which run from the In-Line Tap to the pod and from the In-Line Tap to the Server). Additionally, inserting the In-Line Tap runs the risk of service-disruption. Finally, the thing is flakey, and getting it to capture packets takes coaxing, not all of which is clear to me yet. Bluntly, I spend obscene amounts of time getting this thing to work. And … we have squandered data capturing opportunities because of its flakiness: during the two Sunday night BlueHeat data gathering episodes, the Pod has captured exactly zero useful traces -- all its captures were invalidated by flakiness issues.

Acquiring a reliable Pod would increase the chances of capturing data in high-performance situations.

If we dedicated Taps to popular servers, like Indigo, we would reduce the time & effort needed to perform High-Performance sniffing.[14]

Historical and Intermittent Sniffing

If we want the ability to answer the question "What happened last night?" or if we want to improve our ability to capture intermittent problems, then we need a new tool: Always On Sniffers. Typically, these are Pods glued to gigabytes or terabytes of disk, capturing everything they see and saving it to a rolling, disk-based buffer.

[14] Of course, there is a decision hidden here: we would need to pick Aggregation Taps if we want to perform Probe-based High-Performance Sniffing … but In-Line Taps if we wanted to perform In-Line Probe-based sniffing.

The catch, of course, is figuring out where to install these things -- they aren't portable. Likely locations, to my way of thinking would be the following: ga-116, j4-401, and cf-114.

Secondary locations would be gb-113 and df-120.

Acquiring one or more Always On Sniffers would allow us to tackle Historical Situations and reduce MTTR for Intermittent Situations … assuming, of course, that the relevant traffic traversed a location served by an Always On Sniffer.

Fibre Channel Sniffing

I propose that we ignore Fibre Channel sniffing for the time being, because our Fibre Channel installation is small -- confined to a single device (Indigo) supported by a single vendor (BlueArc). If and when we expand our Fibre Channel network to include more than one box and/or multiple vendors, then I propose that we revisit this choice.

Here is rough Fibre Channel pricing.

In-Line Probe (Probe, Tap, Software):	16K
Cisco Ethernet/Fibre Channel converter:	3K
NetWisdom, Low-End:	30K

Do nothing.

PRICING

Here is a rough feel for pricing.[15] Remember that commercial products tend to cost ~10% per year in maintenance fees.

Probes

Probes vary from .8K - 2K
Pods vary from ~20K to ~45K.

Always On

NETWORK GENERAL

InfiniStream, 300GB disk, 18K
InfiniStream, 1TB disk, 35-45K
InfiniStream, 4TB disk, ~100K

[15] Consider figuring out how to build our own Pods, using Xyratex or Endace cards and Ethereal. Consider watching eBay for deals.

Taps

In-Line	1.5K
Aggregation	2K - 3K
EtherChannel Aggregation	4K

Visualization Software[16]

NETWORK GENERAL
Visualizer starts at 35K

WHAT DO WE DO NEXT?

Having outlined what we could do in Synthesis … what should we actually do?

This is a hard question to answer, because we have no easy to quantify the cost of downtime. If I could quantify how much a given tool would reduce MTTR and if knew that we lost a $1000/hour say, when Indigo was down … then I could start quantifying this decision. But I don't know either parameter.

IMHO, the most frequent strains on our analysis tool set to date are as follows, in order of frequency:

- Sniffing on either side of the firewalls
- Sniffing at Client *and* Server locations simultaneously

In terms of visibility, a third category has arisen recently, High Performance Sniffing, wrt to the BlueHeat project.

Here is a stab at recommendations.

Caveats

Notice that I've focused my attention almost exclusively on packet capture and packet analysis tools … I haven't begun to touch the larger tool set, including such things as graphing tools, real-time SNMP counter grabbers, and a host of others.

Note that I have not done a complete product evaluation … I'm still in the middle of an early eval cycle sufficient to inform our budgeting process but not detailed enough to inform an actual purchase.

[16] Note that one must be a server with plenty of disk and CPU in order to run these packages.

Insurance

If we decide that we are poor right now, then we live with what we have: it allows us to tackle all the Use Cases except for Historical, Intermittent, and Fibre Channel, and that's a pretty robust place to be. As insurance, we set aside money to hire Mike Pennachi to tackle difficult problems for us. Mike charges ~1K/day for this Critical Problem Resolution service. If we don't hire Mike in a given year, then we can redirect the dollars somewhere else. I will arbitrarily declare that Mike can solve any problem given a week.

5K

Add a Probe

We buy a third Probe. If we're feeling particularly flush, we add an Aggregation Tap, allowing for In-Line sniffing, albeit not at line-rates.

2K - 6K

More Probes

We dedicate a Probe to cf-114 (aka Pond) and a Probe to ga-116 (aka scca-Pond). Each would contain multiple NICs, allowing us to sniff outside the firewall and inside the firewall (and in the SCCA's case, off ga-a-rtr, giving us visibility into intra-building traffic). If we're feeling particularly flush, we add Aggregation Taps, to allow us to sniff in-line between the IPS and the Firewalls.

3K - 10K

Prepare for Future High Bandwidth Situations

Buy a more reliable Pod.

20K-45K

Tackle Intermittent Situations and Historical Situations

We buy one or more Always On Sniffers.

20K - 200K