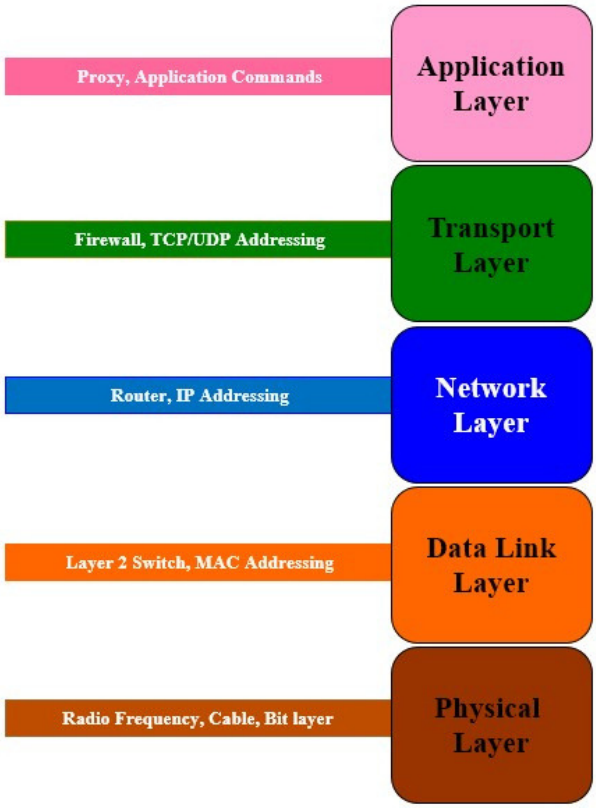
NETBEEZ

Operational Guidelines and Tools for Network Testing

When it comes to network testing, the most efficient methodology and tools is the formula for success. The issues technicians encounter are losing focus or skipping steps when under pressure.

In this document, I will help build your toolbox and methodology. To keep this document as concise as possible, I took a high-level approach to better explain every tool and layer along the way.

Seeing something graphical helps visualize the relationship between the tools and the various layers.



These five layers from the OSI Model is the best way to visualize what areas to cover.

Each layer correlates to a network function, not necessarily a physical device. In most cases, multiple layers can cover a single physical device.

For example, a router with NAT configured, will cover the Physical, Data Link, Network and Transport Layers.

Regardless of what layers are covered by that device, you should still ensure that every layer is tested and monitored from the bottom up.

Layer 1:Cable Testers and Spectrum Analyzers

When I present, I sarcastically say, “If it can electrocute, blind, burn or hang you, its layer one". In some cases, this is the most straightforward layer since most of this layer is something you can physically touch, feel or see.

At this layer, I am referring to cable testers and/or spectrum analyzers. A time-domain reflectometer (TDR) is an electronic instrument to characterize and locate faults in twisted pair wire. It can also be used to locate discontinuities along an electrical path. The equivalent device for optical fiber is referred as an optical time-domain reflectometer.

There is a difference between cable certification tools and testing/qualification tools so take the time to figure out what you have.

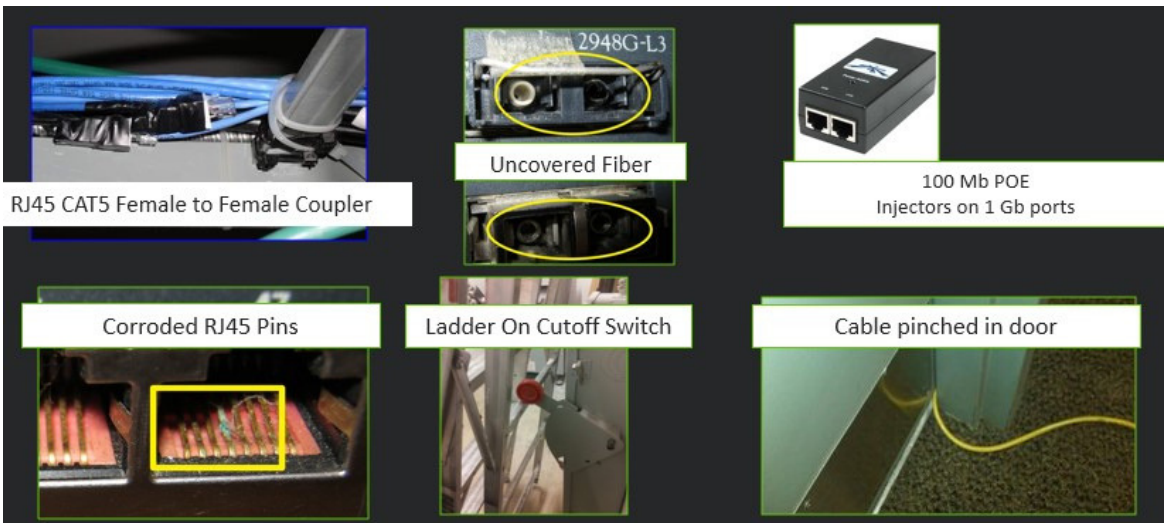
Check your network equipment documentation since some switches can perform simple TDR tests.

The image below is an example from a Cisco 3750 ‘show cable-diagnostic” output.

```
Dexter# test cable-diagnostics tdr int g2/0/1
TDR test started on interface Gi2/0/1
A TDR test can take a few seconds to run on an interface
Use 'show cable-diagnostics tdr' to read the TDR results.
Dexter# sh cable-diagnostics tdr int g2/0/1
TDR test last run on: July 11 09:27:40
```

Interface	Speed	Local pair	Pair length	Remote pair	Pair status
Gi2/0/1	1000M	Pair A	42 +/- 10 meters	Pair A	Normal
		Pair B	42 +/- 10 meters	Pair B	Normal
		Pair C	42 +/- 10 meters	Pair C	Normal
		Pair D	42 +/- 10 meters	Pair D	Normal

A physical inspection is also recommended to ensure that the cable isn’t physically damaged. Below are various examples of problems I have come across in the field to look out for.



It also pays to use your network monitoring system or equipment CLI commands to help identify ports that may have physical layer issues. This technique is invaluable when working on remote sites.

The output below is an example from the “show interface count error”, a Cisco 3750.

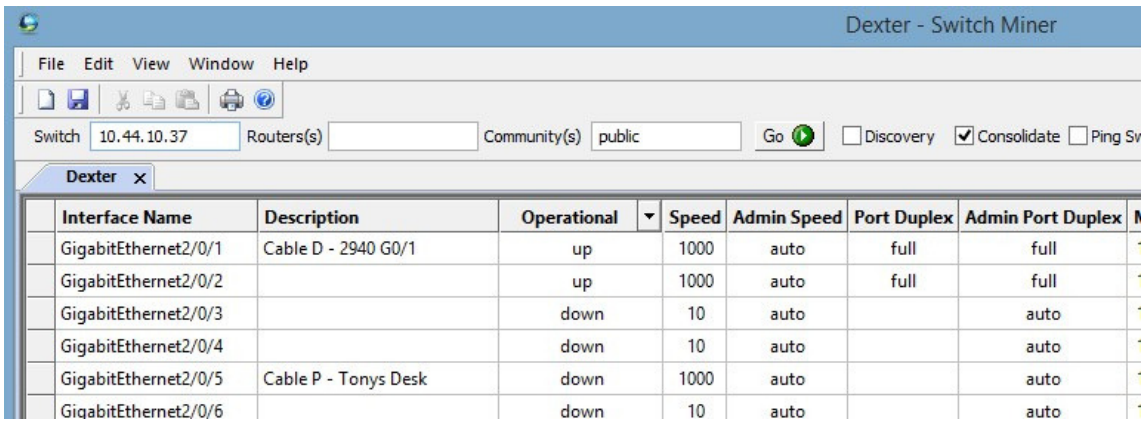
```
Dexter# show interface count error

Port      Single-Col  Multi-Col  Late-Col  Excess-Col  Carri-Sen  Runts  Giants
Gi2/0/1        0          0          0          0          0        0      0
Gi2/0/2        0          0          0          0          0        0      0
Gi2/0/3        0          0          0          0          0        0      0
Gi2/0/4        0          0          0          0          0        0      0
Gi2/0/5        0          0          0          0          0        0      0
Gi2/0/6        0          0          0          0          0        0      0
Gi2/0/7        0          0          0          0          0        0      0
Gi2/0/8    14688       9184          0          0          0        0      0
Gi2/0/9        0          0          0          0          0        0      1
Gi2/0/10       0          0          0          0          0        0      0
Gi2/0/11     801      28636      13966        193          0        0      0
Gi2/0/12       0          0          0          0          0        0      0
Gi2/0/13       0          0          0          0          0        0      0
Gi2/0/14     204         61          0          0          0        0      0
```

Lastly, don’t forget about speed/duplex mismatches. It is important to check your port settings are configured for versus what the actual speed/duplex is. Switchminer (http://switchminer.sourceforge.net) is a great free open source tool to will query your switch and display your speed, duplex, and errors.

Layer 2: MAC Adresses

This layer involves devices and tools that recognize a MAC address. A big part of this is identifying what port devices reside on.

SwitchMiner and similar applications can produce a table or report of MAC addresses and their corresponding port location.



Wireshark (http://www.wireshark.org) or your favorite protocol analyzer is a great way to identify MAC addresses and any layer 2 related issues.

Things to look for range from security like ARP flooding, spoofing and man in the middle attacks on performance related issues like load balancing, flooding and Spanning Tree.

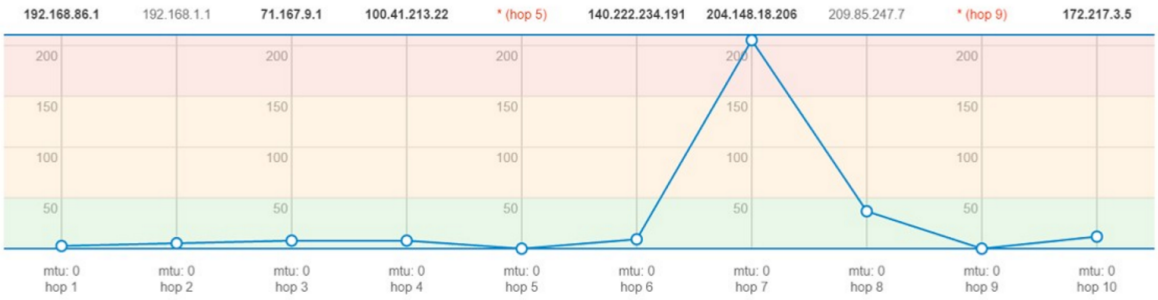
Layer 3: Ping and Traceroute

This is the most well-referenced layer since it covers tools like ping and traceroute. This layer can get quite involved so let's start with the basics.

Good old ping is used to test reachability as well as response time. It is important to note that ping uses ICMP and will typically default to a small payload and allows IP fragmentation. When troubleshooting, you should modify the default payload size to match your application packet size and consider disabling IP fragmentation so its treated more like a TCP packet since most client operating systems do not allow IP fragmentation for TCP.

Traceroute is a similar utility that reports back the IP address and name of all layer 3 devices along the path to the destination device by increasing the IP time to live. The fragmentation tip mentioned earlier also applies to traceroute.

You should have a TCP or UDP ping or traceroute tool, like NetBeez, which provides a browser-based interface for both of these tests.



This is where things can get confusing since you are referencing a layer 4 port number but testing a layer 3 route.

Having a layer 3 tool to test reachability and routes (where possible), where you can modify the payload size and IP fragmentation is very helpful. An added bonus would be a scheduling and reporting feature as well as alerts (syslog, email or log) when thresholds are exceeded.

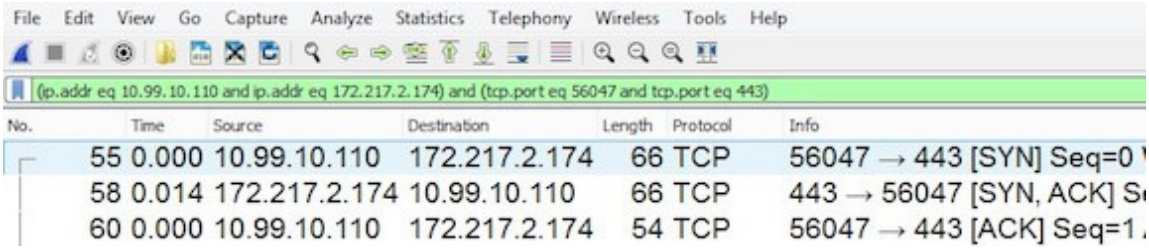
Layer 4: TCP and UDP

This layer involves TCP or UDP port numbers which imply that you need to know what port numbers your application uses. In some cases, it will be obvious. For example, a web-based application will use TCP port 80 or 443 by default. If using Chrome and QUIC it could be UDP port number 443 when communicating to various Google sites.

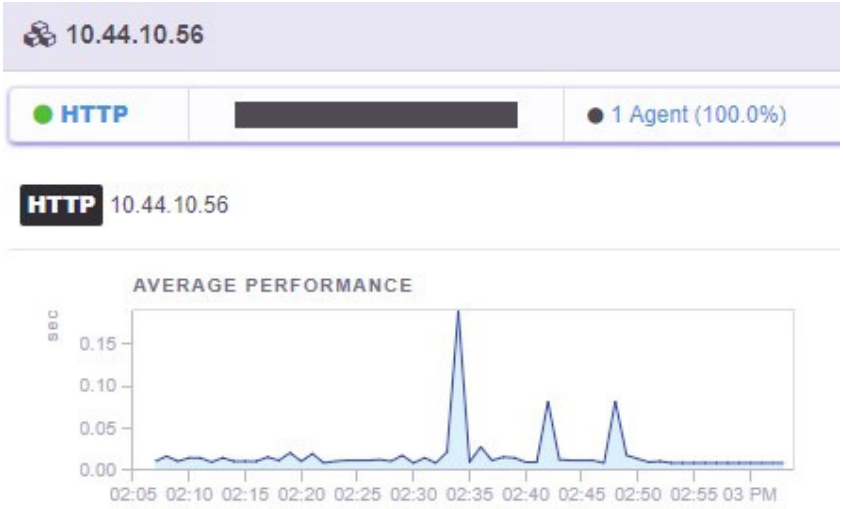
Try out the windows netstat –b command from an elevated command prompt. Currports (http://www.nirsoft.net/utils/cports.html) is a free, portable, Windows GUI based application that will provide the same information as netstat –b with extra features.

At Layer 4 you need tools or methodologies that measure the time between the TCP SYN and SYN ACK. This is also referred to as the 'TCP connect time'.

The screenshot below is from Wireshark and illustrates a TCP connect time of 14 ms.



As mentioned with traceroute and ping, having a scheduling option would be very helpful. NetBeez provides reporting and scheduling for their HTTP checks. The screenshot below is from NetBeez's dashboard.



Layer 7: Performance Measurement

This is the application layer where we focus on measuring performance and other application related issues. The most straightforward way of doing this is to capture the packets of a real client running a real application on your real network. The reason why I emphasize the word 'real' is because one option at this layer is to use products that simulate or model your applications. While there is isn't anything wrong with this I prefer real data.

After you capture your packets, analyze the command and response delta times. One technique to make is easier is to filter on the ip/tcp port number (or conversation) combination.

In the screenshot below you can see the response took 68 ms to reply (65 + 3).

```
 9 0.000   10.44.10.176    74.208.236.106  576 HTTP GET /networking/networking.htm HTTP/1.1
15 0.065   74.208.236.106 10.44.10.176      60 TCP   80 → 60499 [ACK] Seq=1 Ack=523 Win=30272 Len=0
16 0.003   74.208.236.106 10.44.10.176     213 HTTP HTTP/1.1 304 Not Modified
```

You can include performance at this layer (as well as layer 1).

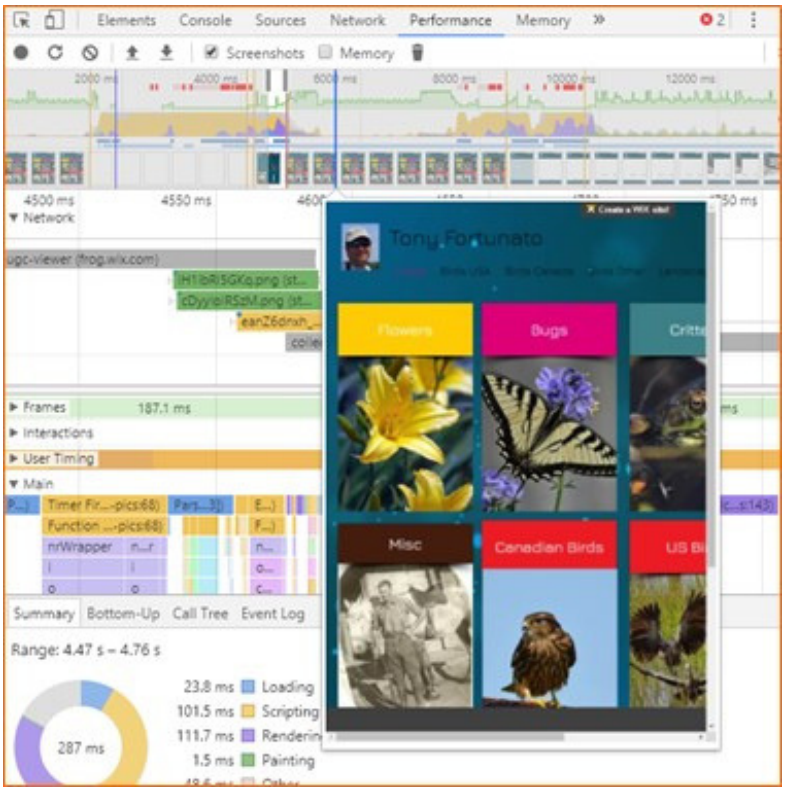
If you use Microsoft Message Analyzer, you can also see which service is associated to a packet.

128		2017-10-15T14:52:06.2478299	vncviewer.exe	10.44.10.186	10.44.10.94
130		2017-10-15T14:52:06.4526213	System	10.44.10.94	10.44.10.186
131	ℹ	2017-10-15T14:52:09.1422643	iexplore.exe	10.44.10.186	108.174.10.10
132	ℹ	2017-10-15T14:52:09.1447934	iexplore.exe	10.44.10.186	74.208.236.106
133	ℹ	2017-10-15T14:52:09.1448330	iexplore.exe	10.44.10.186	74.208.236.106

This is where many of the Application Performance Measurement tools come in as well as some of the protocol analyzers that provide advanced reporting.

One example of automation tools that can measure response time is Apptimer (https://www.passmark.com/products/apptimer.htm).

The other powerful web analysis tool is using the Developer Tools in Internet Explorer or Chrome.



Tony Fortunato is a Sr. Network Performance Specialist with The Technology Firm who has been, designing, implementing, and troubleshooting networks since 1989 as well as customized training. Tony can be reached via www.thetechfirm.com