SOLARWINDS WHITEPAPER

Using Packet Analysis for Quality of Experience Monitoring

By: James H. Baxter

Performance Analyst / WCNA

getpackets / PacketIQ® Inc.

Share:

solarwinds

Unexpected Simplicity™

About the Author

James H Baxter is a Performance Analyst for getpackets and the president and CEO of PacketIQ Inc. With over 30 years of experience in the IT industry, his technical background includes electronics, RF, satellite, LAN/WAN & voice design, network management, speech technologies, Java/.NET programming.

For most of the last 18 years, James has been working specifically with network and application performance issues. He is a Wireshark ® Certified Network Analyst (WCNA), an active member of the IEEE, CMG, and ACM, and follows advances in Artificial Intelligence.

Introduction

In this whitepaper, I'm going to discuss what Packet Analysis is, some of the useful information it can provide, and how this info can be used to monitor the application end-user experience.

Packet Analysis

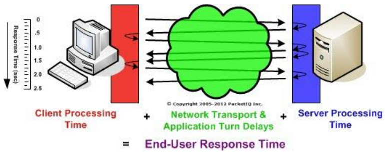
The IT industry is increasingly recognizing and leveraging the value and utility of packet-level analysis—also called Deep Packet Inspection or just Packet Analysis—for quickly and accurately identifying the true source and nature of network and application reliability and performance problems.

Packet analysis involves 'capturing' (making a copy of) and inspecting network packets that flow between client and server devices. Typically, this is accomplished with a tool commonly referred to as a 'Sniffer,' which is the name of one of the first industry standard tools designed for packet analysis.

More recently, an open-source software application called 'Wireshark®' (formerly known as 'Ethereal') has become the leading tool for manual packet analysis. Wireshark can be installed on a workstation or laptop. It utilizes a promiscuous (capture everything) mode driver with a built-in network interface card, making this a very capable tool. However, this solution is typically moved around and used on an as-needed basis. There's also a lot of skill required to properly configure the software, perform captures, and analyze/interpret packet flows.

A number of vendors offer specialized appliances to perform high-throughput and/or deep packet inspection at the enterprise level. However, these tools can be very expensive to deploy, especially when deploying them to a number of locations with the expectation of providing optimal coverage.

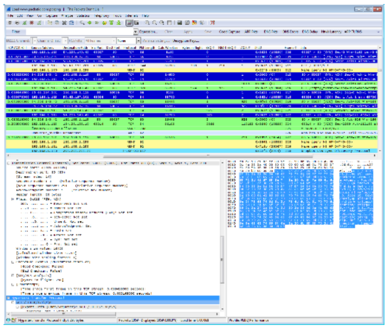
The SolarWinds® Quality of Experience (QoE) console, which has been added to Network Performance Monitor, leverages packet analysis to offer an excellent solution for dedicated, full-time monitoring of critical network and server performance as well as traffic type and volume distributions at an affordable price.



Primary factors that affect end-user response time, or Quality of Experience, include:

- *Client Processing Time*
- *Server Processing Time*
- *Network Transport Delay*
- *Application Turn Delay*

The most significant of these are Server Processing and Network related delays.



Wireshark capture screen

What Packet Analysis Can Provide

In addition to transporting data between clients and servers, modern networking protocols such as TCP/IP are tasked with ensuring the reliable delivery of packets, minimizing packet loss, and maintaining data flow controls in order to optimize network throughput—most importantly when dealing with congested networks. By inspecting packet flows and protocol parameters, useful information about network performance can be extracted.

Packet analysis can also identify all types and relative volumes of application traffic flowing over a network based on the host IP addresses, ports, and protocols in use.

By inspecting and interpreting network data flows at the packet level, a wealth of performance related information can be gleaned. The following sections discuss and illustrate some of the most useful of the possibilities.

Network Response Time

Network response time—or network path latency, as it's also known—is a measure of the amount of time required for a packet to travel across a network path from sender to receiver. When network path latencies occur, application performance is often adversely affected.

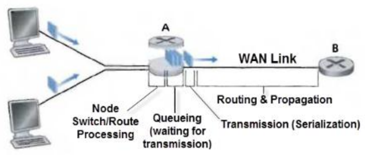
Causes of High Network Response Times

Four major factors affect network path latency:

- Speed-of-Light Propagation Delay
- Network Routing/Geographical Distance
- Serialization Delay across Wide Area Network (WAN) Links
- Queuing Delays in Network Devices

Speed-of-Light Propagation Delay

Electrical signals travel through the vacuum of space at the speed of light—about 186,000 miles per second, or 299,792 kilometers per second, or 671 million miles per hour, depending on your preference. Despite this almost unimaginable speed, it still takes a finite amount of time for an electrical signal to travel across distances common to our frame of reference. The



Causes of Network Transport Delay

The four major causes of network path transport delay include propagation, routing, serialization, and queuing delays.

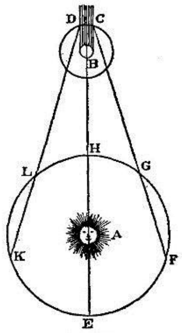


Fig. 70.

The first quantitative estimate of the speed of light was made in 1676 by Ole Christensen Rømer from the observation that the periods of Jupiter's moon Io appeared to be shorter when the Earth was approaching Jupiter than when receding from it.

Light takes 8 minutes and 19 seconds to travel from the Sun to the Earth.

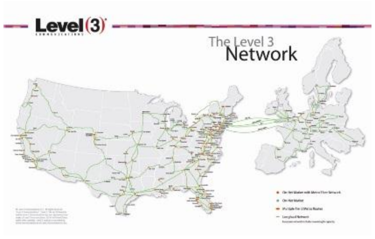
-- Wikipedia

straight-line distance between New York to Chicago, for example, is about 713 miles—it takes light about 3.8 milliseconds (ms) to travel that far.

However, it takes longer for an electrical signal to propagate through physical media, such as the copper and fiber optic cables commonly used in telecommunications that connect devices together across both long and short distances. Electrical signals travel through copper and fiber at only ~66% of light speed—so it would take about 5.8 ms for a signal to get from New York to Chicago in this scenario. And, of course, the longer the distance between sender and receiver, the longer it takes for the signals to arrive.

Network Routing/Geographical Distance

The propagation delay example above represented an ideal, straight-line distance. In the real world, signals routed between New York and Chicago would most likely go through several major switching centers in locations not directly between the two end-points. In addition, the cabling between these switching centers would by necessity be laid along opportunistic paths beside railroads and major highways and other right-of-way corridors. The end result being a dog-leg physical route that can end up being a great deal longer than the ideal straight-line path.



Level 3 Network routes in the US – Europe - UK

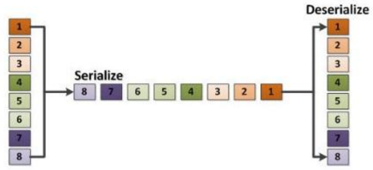
This additional distance can add a considerable amount of network path delay. It's not unusual to see Round Trip Time (RTT) (Client to Server and Return) network latencies on the order of 50-100 ms or more over typical US or Europe/UK network paths, and much higher than that over trans-global distances or satellite hops.

Serialization Delay Across Wide Area Network (WAN) Links

Although network access and Internet links out of and between a company's major data centers are usually designed for high bandwidths (600+ Mbps) to accommodate the traffic demand, WAN links to distribution centers and especially branch offices can be much smaller—ranging from 512 Kbps to 45 or 100 Mbps.

At lower link speeds, a significant amount of time can be consumed from 'clocking' the number of data bits contained in a typical network packet onto these lower bandwidth portions of a network path. Consider the following example:

A typical large network packet can be 1476 bytes long



Data is inserted bit-by-bit onto network links at link speed rates. The slower the link, the longer it takes to clock a given amount of data onto the link.

http://e2e.ti.com/blogs_/b/analogwire/arc

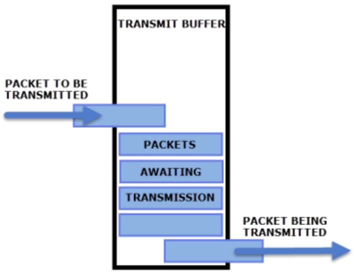
1476 x 8 bits/byte = 11,808 bits

For a branch office serviced by a T-1 (1,536 Kbps available bandwidth for data), it takes:

11,808 bits/1,536,000 bits/sec = .00768 or 7.68 ms

That is the time taken just to ‘clock’ or ‘serialize’ the packet data through the WAN interface and onto the link (one bit at a time, at the link speed rate) so that the data can be transmitted to the other end. The same packet would take 23 ms to serialize onto a 512 Kbps link, but only 1.2 ms to serialize onto a 10 Mbps link. There are increasingly lesser amounts of time required for higher speed links—so these delays become relatively insignificant for high-speed links.

Serialization delays obviously contribute to overall network delay. Aside from using higher speed links (which becomes increasingly expensive), there is little that can be done to reduce the effects of this type of delay aside from using packet data compression, caching, and other techniques (which are used in WAN optimization appliances) to reduce the overall number of packets needed to service requests and/or deliver data.



Transmit buffer queuing

Queuing Delays in Network Devices

Small-node processing and switching delays occur as packets traverse various switches and routers along a network path. However, delay time is relatively insignificant compared to the time spent when packets are held up in router buffers awaiting their turn to be transmitted across network links—especially slower WAN links.

This type of delay is closely related to the serialization delay described in the previous section. Packets stack up and wait in transmit buffer queues while packets ahead of them are serialized onto slower WAN links (which can increase when links get busy). In addition, a packet’s time in the queue can also be affected by Quality of Service (QoS) policies that may be in place to help ensure network performance is optimized for critical applications.

A short detour to discuss time-sensitive applications, QoS, and the reasons for its use is probably in order.

Network Effects on Sensitive Applications

Some applications such as voice and video-conferencing are sensitive to packet loss, delay, and 'jitter' (variances in the delivery time between sequential packets).

Packet loss and jitter can affect the overall quality of a voice call (causing 'stuttering' and strange tonal effects), while high overall path delays can increase the noticeable effects of echo and hinder speaker interactions due to 'talk-over'.

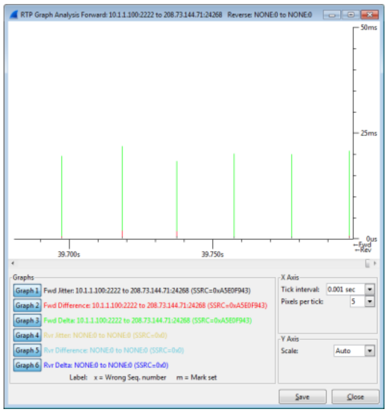
Packet loss and high path delays can have even more profound effects on video conferencing (aka 'telepresence') systems. Packet loss can cause "blockiness" and/or jerkiness of the video, as well as audio drop-outs. High network latency can cause loss of lip-synchronization because the relatively small voice packets may be delivered in a different timeframe than the larger voice content packets. Therefore, it's important that voice and video packets flow across the network on a consistent, reliable basis.

When network congestion from short-term peaks in traffic volume occur, QoS policies can help make sure time and delay sensitive applications continue to perform as expected.

Quality of Service Policies

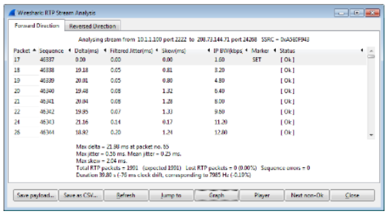
Network administrators can configure QoS policies in switches and routers to mark packets from different types of applications with a 'Differentiated Services Code Point' (DSCP) value that identifies their relative priority. As each packet traverses the network devices along a path they may be handled differently based on their DSCP codes—this is called Per-Hop Behavior (PHB).

When a network link—typically a WAN link—becomes congested, packets will receive priority based on the class of DSCP codes they've been assigned. High priority classes of DSCP codes include the Expedited Forwarding (EF) or Assured Forwarding (AF) codes that may be applied to voice and multimedia traffic. Other, less time-sensitive packets (such as email traffic) are marked as a lower priority and may sit in different router transmit queues, for longer periods of time, awaiting their turn to be sent. In cases of extreme congestion, some packets may be dropped altogether, resulting in packet loss.

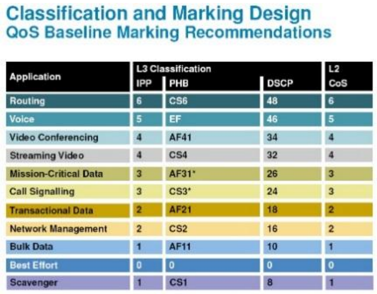

RTP Graph Analysis Forward: 10.1.1.100:2222 to 208.73.144.71:24268 Reverse: NONE:0 to NONE:0

Graph 1 Fwd Jitter: 10.1.1.100:2222 to 208.73.144.71:24268 (SSRC=0xA5E0F943)
Graph 2 Fwd Difference: 10.1.1.100:2222 to 208.73.144.71:24268 (SSRC=0xA5E0F943)
Graph 3 Fwd Delta: 10.1.1.100:2222 to 208.73.144.71:24268 (SSRC=0xA5E0F943)
Graph 4 Rvr Jitter: NONE:0 to NONE:0 (SSRC=0x0)
Graph 5 Rvr Difference: NONE:0 to NONE:0 (SSRC=0x0)
Graph 6 Rvr Delta: NONE:0 to NONE:0 (SSRC=0x0)
Label: x = Wrong Seq. number m = Mark set

VoIP / RTP Inter-Packet Time & Jitter Analysis


Wireshark: RTP Stream Analysis

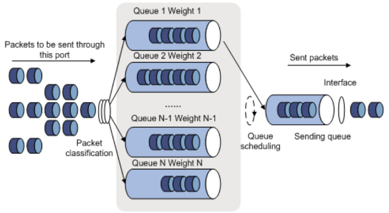
```
Analysing stream from 10.1.1.100 port 2222 to 208.73.144.71 port 24268 SSRC = 0xA5E0F943

Packet  Sequence  Delta(ms)  Filtered Jitter(ms)  Skew(ms)  IP BW(kbps)  Marker  Status
17      46337     0.00       0.00                 0.00      1.60         SET     [ Ok ]
18      46338     19.18      0.05                 0.81      3.20                 [ Ok ]
19      46339     20.01      0.05                 0.80      4.80                 [ Ok ]
20      46340     19.48      0.08                 1.32      6.40                 [ Ok ]
21      46341     20.04      0.08                 1.28      8.00                 [ Ok ]
22      46342     19.95      0.07                 1.33      9.60                 [ Ok ]
24      46343     21.16      0.14                 0.17      11.20                [ Ok ]
26      46344     18.92      0.20                 1.24      12.80                [ Ok ]

Max delta = 21.98 ms at packet no. 65
Max jitter = 0.56 ms. Mean jitter = 0.25 ms.
Max skew = 2.04 ms.
Total RTP packets = 1991 (expected 1991) Lost RTP packets = 0 (0.00%) Sequence errors = 0
Duration 39.80 s (-76 ms clock drift, corresponding to 7985 Hz (-0.19%)
```


Classification and Marking Design
QoS Baseline Marking Recommendations

Application	L3 Classification			L2
	IPP	PHB	DSCP	CoS
Routing	6	CS6	48	6
Voice	5	EF	46	5
Video Conferencing	4	AF41	34	4
Streaming Video	4	CS4	32	4
Mission-Critical Data	3	AF31*	26	3
Call Signalling	3	CS3*	24	3
Transactional Data	2	AF21	18	2
Network Management	2	CS2	16	2
Bulk Data	1	AF11	10	1
Best Effort	0	0	0	0
Scavenger	1	CS1	8	1

QoS classifications for various application types

Cisco QoS class at Cisco Live 2009 [QoS on Cisco Switches]



Packet buffer queues

http://www.h3c.com/portal/Products___Solutions/Technology/QoS/Technology_Introduction/200701/195599_57_0.htm

When network congestion occurs, QoS policies help ensure time- and delay-sensitive applications perform as expected and that the most value is extracted from expensive network links. At the same time though, these same policies can negatively affect the performance for lesser priority applications—including user-interactive/transactional applications.

Effects of Network Delay

The accumulative effects of propagation, routing, serialization, and queuing delays can result in relatively sizable total network delays between a Client and Server over typical distances.

TCP/IP protocols work to overcome these delays for bulk data transfers with techniques such as 'sliding windows' wherein multiple packets can be sent and outstanding before they must be 'acknowledged', resulting in a more efficient data transfer.

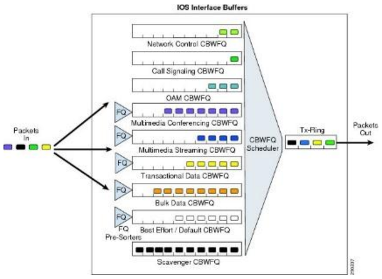
Direct user interactions with a server, however, are adversely affected by higher network delays, making some applications become 'chatty.' This occurs when they utilize a fairly high number of request/response cycles to accomplish a task (called 'Application Turns' or just 'App Turns'). Each of these request/response cycles incur the round-trip network delay time—the total time for high App Turn applications over a high latency path. It's important to note that this time can add up very quickly.

For example, loading a moderately complex Web page that contains a high number of graphic images and multiple CSS and JavaScript files can result in a LOT of requests—one for each file. If there are 35 files required to load one page over a 72 millisecond RTT path, the total time incurred from App Turns alone is:

35 x .075 = **2.625 seconds**

This is *just* for App Turns delay—waiting for requests and responses to traverse the minimum network routing/propagation delay. This delay is in addition to any client and server processing time and other network transport delays (serialization and queuing) incurred to transmit the actual data bytes. It's easy to see how loading a busy Web page over a moderately high latency path can take multiple seconds.

Recent improvements in Web page design and optimization techniques include using more CSS instead of graphics for page formatting, choosing



Simplified diagram illustrating QoS queuing

http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND_40/QoSWAN_40.html

Web browsers will limit the number of simultaneous downloads (connections) per server, as follows:

Internet Explorer® 7 and earlier: *2 simultaneous connections per host/IP*

Internet Explorer 8 and 9: *6 connections per host*

Internet Explorer 10: *8 connections per host*

Internet Explorer 11: *13 connections per host*

Chrome™ 32: *6 connections per host*

Firefox® 26: *6 connections per host.*

See: http://www.browserscope.org/?category=network optimal graphic formats and sizes, and consolidating and compressing CSS and JavaScript files (using Gzip) so that fewer and smaller requests are needed to build a page. In addition, current Web browsers can now employ as much as 6-8 simultaneous connections to a Web server, enabling the completion of multiple requests in a relatively short time frame. However, not all websites utilize these optimizations. Many companies are still locked into older browsers or 'chatty' legacy applications that can't or won't upgrade for some time. In this case, App Turns delays continue to be a significant factor in application performance.

As can be seen from the previous discussions, network delay in all its manifestations is a significant and important factor in overall application performance—and one that certainly warrants monitoring for mission-critical applications.

Network Delay Measurements

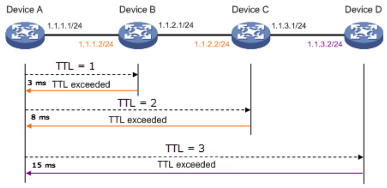
Network delay across a path between a client and server is often measured (by a manual process) using ICMP 'Pings.' It can also be observed and measured at the packet level from the amount of time that transpires between specific packets used in a sequence required for establishing client sessions with an application server. Some background information is helpful in understanding how this is accomplished.

The TCP/IP 3-Way Handshake

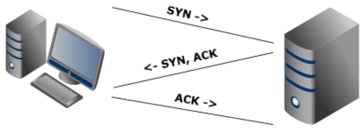
When a client initiates a session with an application server utilizing TCP/IP, a 3-way 'handshake' (**TCP Handshake**) takes place to set up a connection in both directions and exchange parameters and options that establish how data is to be transferred and how error conditions are to be handled.

In the first step of this 3-way handshake, the Client device sends the Server device a 'SYN' (Synchronize) packet that contains an Initial Sequence Number (ISN) that has been randomly generated (to defend against a certain type of hacker attacks).

The server responds with a 'SYN, ACK' packet that contains its own Synchronization number and acknowledges the Client's SYN request. In the last step of the handshake, the Client sends an ACK packet acknowledging the server's SYN connection request. The Client can now send application requests to the server—and in fact, could do in the 3rd (ACK) packet in the sequence.



ICMP Ping packets are used to measure RTT latency, but are also used to perform traceroutes by manipulating the TTL counter to get information for each hop in the path.



TCP 3-Way Handshake

During this handshake, the Client can indicate its ability to support two enhancements to TCP—Selective Acknowledgement (SACK) and Window Scaling—in its initial SYN packet. If the Server also supports these, it's indicated in its SYN, ACK return packet. If these options are not supported by both ends of the session, they cannot be utilized. ***See RFC 2018***

Selective Acknowledgement allows a device to specifically indicate a byte range it's missing due to packet loss—and by association, which bytes it might have received since the missing packet. This allows the sender to retransmit the missing data packets instead of holding up the process until it receives the lost packet(s), thereby requiring all subsequent packets to be re-sent as well.

Window Scaling is a method of indicating that a device's receive buffer can be larger than the maximum of 65,535 bytes it is possible to report using the two-byte window size field in TCP headers. This maximum is indicated by a multiplier value that is to be applied to the reported window size field values.

SACK and Window Scaling features can significantly improve data transport efficiency, are supported by most recent operating systems, and can be verified by inspecting the handshake packets for each session.

Measuring Network Delay from a TCP 3-Way Handshake

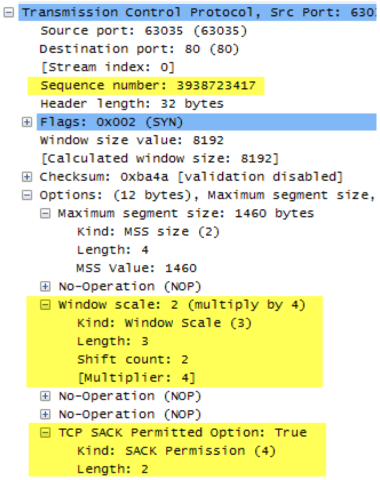
From the Client end of the network path, measuring the time that transpires between the initial Client SYN packet and the Server's SYN, ACK response packet is a reasonably accurate reflection of the Round Trip Time between Client and Server.

If the observation/capture point is at the Server end of the network path, the time between the Server's SYN, ACK packet, and the Client's ACK packet will reflect the network RTT time.

Finally, if the observation point is somewhere in the middle of this path, the network RTT will be reflected as the time between the Client's initial SYN packet and the Client's final ACK packet (3rd packet of the 3-way handshake).

About SEQ and ACK Numbers and Retransmissions

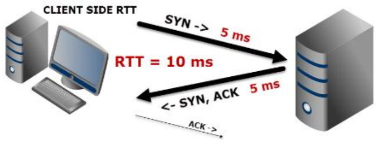
TCP Sequence (SEQ) numbers—along with Acknowledgement (ACK) numbers—are utilized by networked devices to track how much data has been transmitted and received. In addition, they track if any data packets have been lost and which ones need to be re-transmitted. They also



```
⊟ Transmission Control Protocol, Src Port: 630
    Source port: 63035 (63035)
    Destination port: 80 (80)
    [Stream index: 0]
    Sequence number: 3938723417
    Header length: 32 bytes
  ⊞ Flags: 0x002 (SYN)
    Window size value: 8192
    [Calculated window size: 8192]
  ⊞ Checksum: 0xba4a [validation disabled]
  ⊟ Options: (12 bytes), Maximum segment size,
    ⊟ Maximum segment size: 1460 bytes
        Kind: MSS size (2)
        Length: 4
        MSS Value: 1460
    ⊞ No-Operation (NOP)
    ⊟ Window scale: 2 (multiply by 4)
        Kind: Window Scale (3)
        Length: 3
        Shift count: 2
        [Multiplier: 4]
    ⊞ No-Operation (NOP)
    ⊞ No-Operation (NOP)
    ⊟ TCP SACK Permitted Option: True
        Kind: SACK Permission (4)
        Length: 2
```

TCP SYN Packet fields:

- *Sequence number*
- *Window scaling*
- *TCP SACK option*

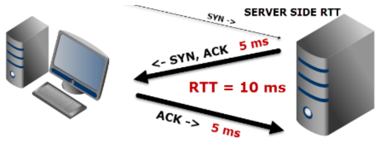


Client-Side Network Round Trip Time measurement

A packet analyzer would show the SYN packet leave the Client, and 10 ms later see the SYN, ACK packet arrive from the Server.

determine the correct order to reassemble packets in case they arrive out of order.

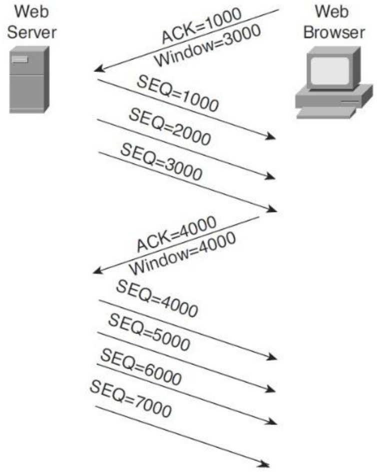
Each time a sending device transmits a packet of data it sets the Sequence number to a value indicating the total number of bytes already sent *before this packet*. The receiving device then indicates that it has received this latest packet by sending an ACK packet back to the sender containing an Acknowledgement number reflecting the number of bytes it has received so far. This in turn indicates the Sequence number it expects to see in the next packet from the sender. The receiver can indicate if any packet in a series of packets gets lost and identify the missing packet. This is accomplished by sending an ACK packet back to the sender with the Acknowledgement number of the last good packet received. As you'll remember, this indicates the Sequence number of the next expected packet. The sender will then re-transmit the missed packet.



A Server-Side RTT measurement over the same path would time the SYN, ACK packet departing and the Client's ACK packet arriving 10 ms later.

It isn't unusual for a relatively few number of packets to be lost during a Client/Server session. The time required for re-transmission of the lost packets isn't significant or noticeable to the end-user. However, higher levels of packet loss from network congestion or faulty network devices *can* have a detrimental and noticeable effect on performance—especially when TCP back-off algorithms kick in and reduce throughput (to accommodate network congestion) on top of packet recovery processes.

You can read RFCs (Request For Comments) 793, 1122, 1323, 2018, and 5681 to get the whole picture on how TCP handles lost packets and retransmissions, congestion, and flow control. You will also get a description of how TCP optimizes performance over higher latency links.



SEQ & ACK number progression with varying Window Sizes

--Wendell Odom

Server Processing Time

Most user-interactive Client applications employ a request/response dialog with a Server, with the expectation that the Server will respond to most requests in a timely fashion—1 or 2 seconds or less.

Client/server-based applications may be hosted on Web servers that rely upon one or more back-end application logic and/or database servers. These servers query the back-end servers for data, process that data, and construct and deliver the response to the client. This may take longer than expected when requests are especially complex, server loading is high, or database queries or application design is inefficient.

As with network delays, server-processing times can be a significant factor in poor application response. But how do you quickly determine which it is? The answer is to monitor both continuously.

Measuring Server Processing Times

Server processing time can be measured at the packet level by observing a specific set of three packets.

When a client makes a request of the server, it will send a packet containing the request and any parameters that may be needed. Typically, the server immediately responds to the request with an 'ACK' packet to let the client know it received the request. Then some period of time will transpire—milliseconds to multiple seconds—while the server processes the request and prepares the response. The next packet seen being sent from the server (to the requesting client) will be the first packet containing the requested data. The time that expires between the Server's ACK packet and the first data delivery packet (aka **'Time to First Byte**) is the Server Processing Time.

So again, the sequence is:

1. Client sends a request
2. Server immediately ACK's the request

some period of time goes by….

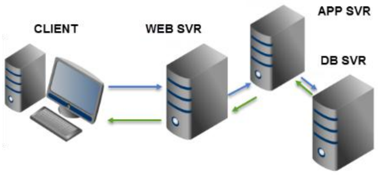
3. Server starts delivering the requested data

Server processing time can therefore be measured by observing the server ACK to delivery packet times within this pattern.

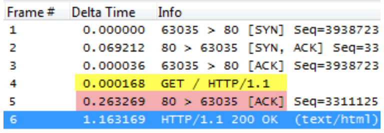
Client Processing Time

Client processing times are not usually a major contributor or area of concern in overall response-time performance analysis, but there are two exceptions that bear mentioning.

Complex client-side application programs (compiled or JavaScript apps) that must do a lot of local processing while also receiving data from a remote server may become too busy to adequately service the received data. An excessive number of other applications (email, Web browser sessions, messaging, etc.) running in the background and consuming excessive processor time (such as virus scanning or Dropbox), may saturate Client



Client – Web Server – App Server – DB Server Request/Response Flows



```
Frame #   Delta Time   Info
1         0.000000     63035 > 80 [SYN] Seq=3938723
2         0.069212     80 > 63035 [SYN, ACK] Seq=33
3         0.000036     63035 > 80 [ACK] Seq=3938723
4         0.000168     GET / HTTP/1.1
5         0.263269     80 > 63035 [ACK] Seq=3311125
6         1.163169     HTTP/1.1 200 OK  (text/html)
```

GET – ACK – RESPONSE sequence. The Server Processing Time was ~1.2 sec

CPU(s) for extended periods of time and affect the Client's ability to adequately service other higher user priority applications.

When a Client gets congested, the receive buffer for a given application that is receiving data may fill up. The reason is, the client can't move the received data out of the buffer over to the application for fast enough processing, or the application itself can't process the data as quickly as it's being received.

Client congestion can obviously affect application performance, and the end-user may not be aware that the poor performance is caused by events on their own workstation.

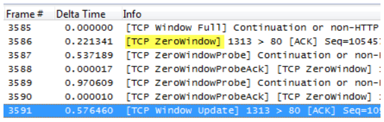
Full receive buffers from congestion isn't limited to just Clients. Servers can and do suffer the same symptoms.

Device congestion can be detected from packet-level analysis by observing the affected receiving device sending a 'Zero Window' packet to the host that is sending data, which tells that host to stop transmitting until it receives a Window Update packet from the congested device indicating that it now has enough buffer space to accommodate another segment of data. In the meantime, the sender will transmit periodic Zero Window Probe packets to the affected device to check if it is ready to receive any new data. These Zero Window Probe packets are sent in time periods that roughly double (~.5 sec, 1 sec, 2 sec, 4 sec, etc.) until either the device's congestion clears up and it sends a Window Update packet, or the session times out and the connection is dropped.

Observation of an excessive number of Zero Window, Zero Window Probe, Zero Window Probe ACKs, and Window Update packets may warrant investigation of the affected device(s).

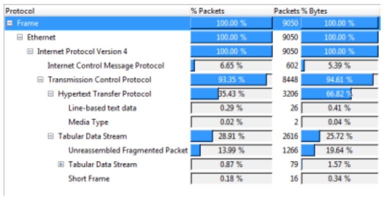
Traffic Distribution Analysis

Packet analysis also allows the categorization of traffic into types based on destination server IP addresses, ports used, and measurement of the total and relative volumes of traffic for each type. This is useful for identifying the volumes of traffic flowing over a network link and/or to specific servers/applications for capacity management purposes. It can also be useful in identifying excessive levels of non-business (social media, external Web surfing, etc.) traffic that may need to be filtered or otherwise eliminated.



```
Frame #  Delta Time  Info
3585     0.000000    [TCP Window Full] Continuation or non-HTTP
3586     0.221341    [TCP ZeroWindow] 1313 > 80 [ACK] Seq=10545
3587     0.537189    [TCP ZeroWindowProbe] Continuation or non-
3588     0.000017    [TCP ZeroWindowProbeAck] [TCP ZeroWindow]
3589     0.970609    [TCP ZeroWindowProbe] Continuation or non-
3590     0.000010    [TCP ZeroWindowProbeAck] [TCP ZeroWindow]
3591     0.576460    [TCP Window Update] 1313 > 80 [ACK] Seq=10
```

ZeroWindow, ZeroWindowProbe, ZeroWindowProbeAck, & Window Update packets



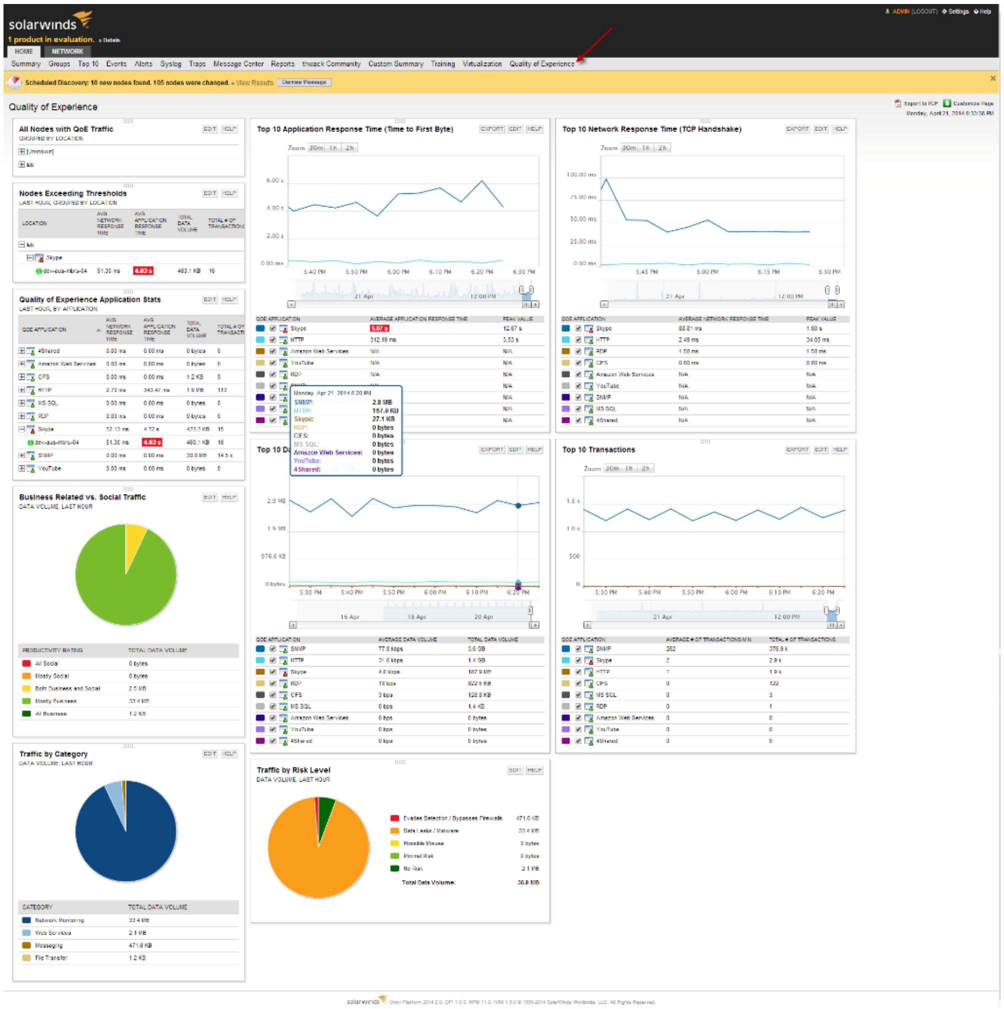
Protocol	% Packets	Packets	% Bytes
⊟ Frame	100.00 %	9050	100.00 %
⊟ Ethernet	100.00 %	9050	100.00 %
⊟ Internet Protocol Version 4	100.00 %	9050	100.00 %
Internet Control Message Protocol	6.65 %	602	5.39 %
⊟ Transmission Control Protocol	93.35 %	8448	94.61 %
⊟ Hypertext Transfer Protocol	35.43 %	3206	66.82 %
Line-based text data	0.29 %	26	0.41 %
Media Type	0.02 %	2	0.04 %
⊟ Tabular Data Stream	28.91 %	2616	25.72 %
Unreassembled Fragmented Packet	13.99 %	1266	19.64 %
⊞ Tabular Data Stream	0.87 %	79	1.57 %
Short Frame	0.18 %	16	0.34 %

Example of Protocol Hierarchy statistics

Quality of Experience Monitoring

The packet analysis techniques discussed in the previous sections outlined a number of opportunities and methodologies for monitoring the most critical factors that may affect end-user application performance, and therefore, their satisfaction.

SolarWinds has developed a solution called Quality of Experience (QoE) that provides full-time, packet analysis-based monitoring and reporting of critical performance factors. This is included in SolarWinds Network Performance Monitor (NPM).



Scheduled Discovery: 10 new nodes found. 105 nodes were changed. » View Results Dismiss Message

Quality of Experience

Monday, April 21, 2014 6:32:38 PM

All Nodes with QoE Traffic
GROUPED BY LOCATION

- ⊞ [Unknown]
- ⊞ lab

Nodes Exceeding Thresholds
LAST HOUR, GROUPED BY LOCATION

LOCATION	AVG NETWORK RESPONSE TIME	AVG APPLICATION RESPONSE TIME	TOTAL DATA VOLUME	TOTAL # OF TRANSACTIONS
⊟ lab				
⊟ Skype				
dev-aus-mbru-04	51.30 ms	4.83 s	480.1 KB	16

Quality of Experience Application Stats
LAST HOUR, BY APPLICATION

QOE APPLICATION	AVG NETWORK RESPONSE TIME	AVG APPLICATION RESPONSE TIME	TOTAL DATA VOLUME	TOTAL # OF TRANSACTI
⊞ 4Shared	0.00 ms	0.00 ms	0 bytes	0
⊞ Amazon Web Services	0.00 ms	0.00 ms	0 bytes	0
⊞ CIFS	0.00 ms	0.00 ms	1.2 KB	5
⊞ HTTP	2.70 ms	343.47 ms	1.9 MB	113
⊞ MS SQL	0.00 ms	0.00 ms	0 bytes	0
⊞ RDP	0.00 ms	0.00 ms	0 bytes	0
⊟ Skype	52.13 ms	4.72 s	423.2 KB	15
dev-aus-mbru-04	51.30 ms	4.83 s	480.1 KB	16
⊞ SNMP	0.00 ms	0.00 ms	30.8 MB	14.5 k
⊞ YouTube	0.00 ms	0.00 ms	0 bytes	0

Top 10 Application Response Time (Time to First Byte)

QOE APPLICATION	AVERAGE APPLICATION RESPONSE TIME	PEAK VALUE
☑ Skype	5.07 s	12.67 s
☑ HTTP	312.19 ms	3.53 s
☑ Amazon Web Services	N/A	N/A
☑ YouTube	N/A	N/A
☑ RDP	N/A	N/A
☑ SNMP	N/A	N/A
☑ [illegible]	[illegible]	N/A
☑ [illegible]	[illegible]	N/A
☑ [illegible]	[illegible]	N/A

Monday, Apr 21, 2014 6:20 PM
SNMP: 2.8 MB
HTTP: 167.9 KB
Skype: 27.1 KB
RDP: 0 bytes
CIFS: 0 bytes
MS SQL: 0 bytes
Amazon Web Services: 0 bytes
YouTube: 0 bytes
4Shared: 0 bytes

Top 10 Network Response Time (TCP Handshake)

QOE APPLICATION	AVERAGE NETWORK RESPONSE TIME	PEAK VALUE
☑ Skype	88.81 ms	1.60 s
☑ HTTP	2.49 ms	34.05 ms
☑ RDP	1.50 ms	1.50 ms
☑ CIFS	0.80 ms	0.80 ms
☑ Amazon Web Services	N/A	N/A
☑ YouTube	N/A	N/A
☑ SNMP	N/A	N/A
☑ MS SQL	N/A	N/A
☑ 4Shared	N/A	N/A

Business Related vs. Social Traffic
DATA VOLUME, LAST HOUR

PRODUCTIVITY RATING	TOTAL DATA VOLUME
🟥 All Social	0 bytes
🟧 Mostly Social	0 bytes
🟨 Both Business and Social	2.5 MB
🟩 Mostly Business	33.4 MB
🟩 All Business	1.2 KB

Top 10 D

QOE APPLICATION	AVERAGE DATA VOLUME	TOTAL DATA VOLUME
☑ SNMP	77.8 kbps	3.6 GB
☑ HTTP	31.6 kbps	1.4 GB
☑ Skype	4.0 kbps	187.9 MB
☑ RDP	18 bps	822.5 KB
☑ CIFS	3 bps	128.8 KB
☑ MS SQL	0 bps	1.4 KB
☑ Amazon Web Services	0 bps	0 bytes
☑ YouTube	0 bps	0 bytes
☑ 4Shared	0 bps	0 bytes

Top 10 Transactions

QOE APPLICATION	AVERAGE # OF TRANSACTIONS/MIN	TOTAL # OF TRANSACTIONS
☑ SNMP	262	376.9 k
☑ Skype	2	2.9 k
☑ HTTP	1	1.9 k
☑ CIFS	0	122
☑ MS SQL	0	3
☑ RDP	0	1
☑ Amazon Web Services	0	0
☑ YouTube	0	0
☑ 4Shared	0	0

Traffic by Category
DATA VOLUME, LAST HOUR

CATEGORY	TOTAL DATA VOLUME
Network Monitoring	33.4 MB
Web Services	2.1 MB
Messaging	471.6 KB
File Transfer	1.2 KB

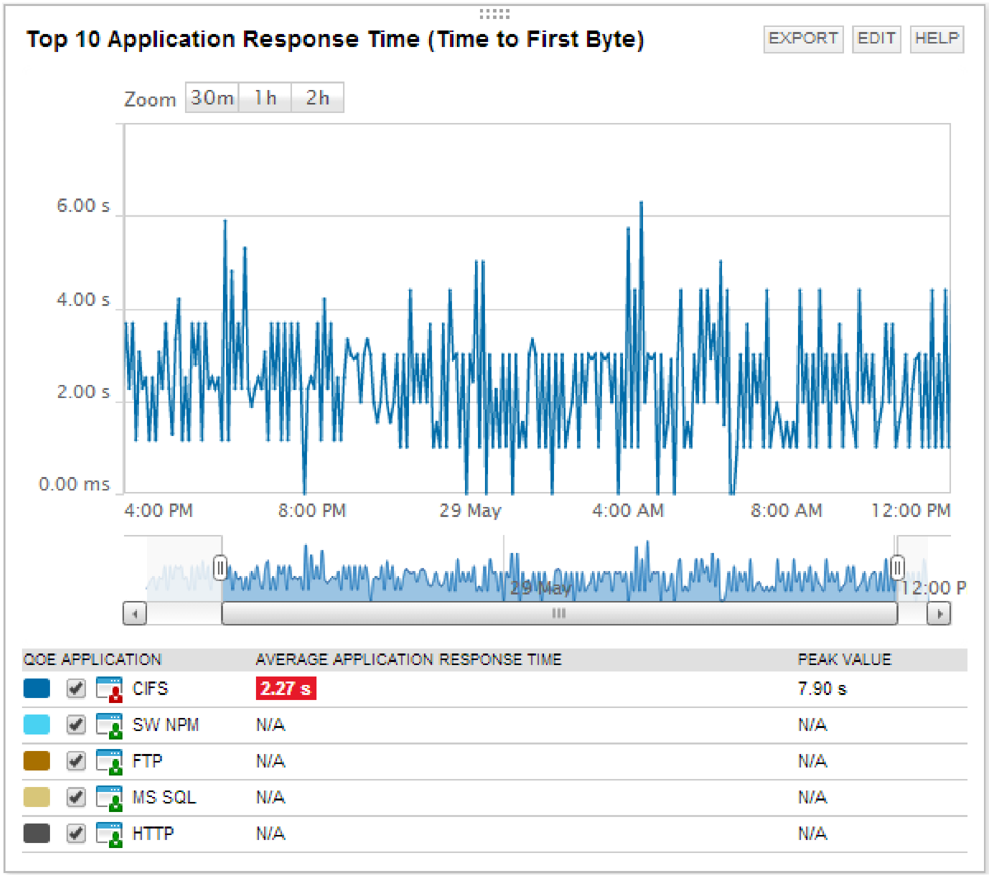
Traffic by Risk Level
DATA VOLUME, LAST HOUR

🟥 Evades Detection / Bypasses Firewalls	471.6 KB
🟧 Data Leaks / Malware	33.4 MB
🟨 Possible Misuse	0 bytes
🟩 Minimal Risk	0 bytes
🟩 No Risk	2.1 MB
Total Data Volume:	36.8 MB

solarwinds Orion Platform 2014.2.0, DPI 1.0.0, NPM 11.0, IVIM 1.9.0 © 1995-2014 SolarWinds Worldwide, LLC. All Rights Reserved.

The SolarWinds Quality of Experience Dashboard

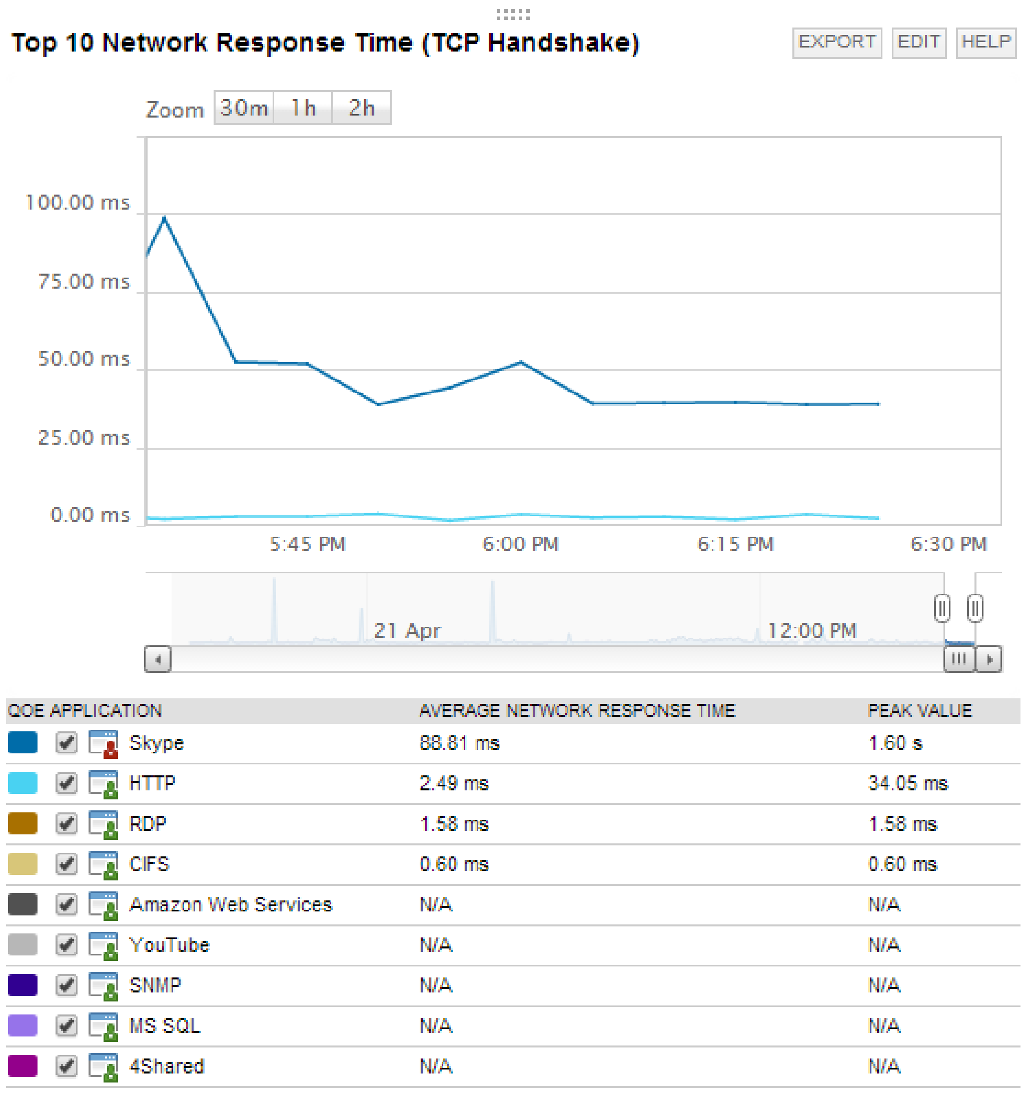
The SolarWinds Quality of Experience dashboard features a quick-glance summary of a variety of network and application performance metrics for all of the configured applications being monitored by QoE Collectors (aka 'sensors').



Top 10 Application Response Time (Time to First Byte)

QOE APPLICATION				AVERAGE APPLICATION RESPONSE TIME	PEAK VALUE
🟦	☑		CIFS	2.27 s	7.90 s
🟦	☑		SW NPM	N/A	N/A
🟫	☑		FTP	N/A	N/A
🟨	☑		MS SQL	N/A	N/A
⬛	☑		HTTP	N/A	N/A

Top 10 Application Response Time performance metrics provide quick notification of app performance issues



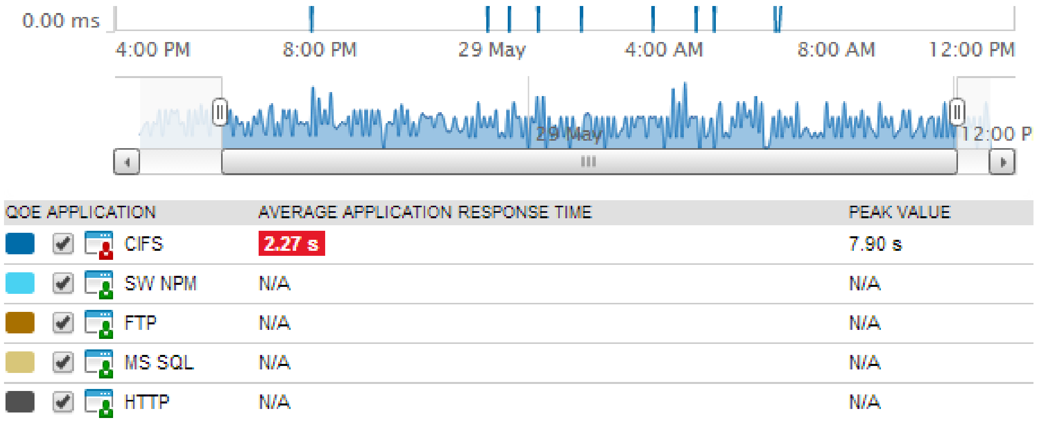
		QOE APPLICATION	AVERAGE NETWORK RESPONSE TIME	PEAK VALUE
🟦	☑	Skype	88.81 ms	1.60 s
🟦	☑	HTTP	2.49 ms	34.05 ms
🟫	☑	RDP	1.58 ms	1.58 ms
🟨	☑	CIFS	0.60 ms	0.60 ms
⬛	☑	Amazon Web Services	N/A	N/A
⬜	☑	YouTube	N/A	N/A
🟪	☑	SNMP	N/A	N/A
🟪	☑	MS SQL	N/A	N/A
🟪	☑	4Shared	N/A	N/A

Top 10 Network Response Time performance metrics provide quick identification of network latency issues

This system processes and reports on performance metrics derived from packet analysis data collected from capture sensors. These capture sensors can be deployed directly on application servers or dedicated collection devices connected to network switch port SPAN/Mirror ports. They provide full-time monitoring of network and application response times for a multitude of pre-configured or custom applications, as well as reporting on overall transaction rates and data volumes against several traffic grouping categories. A key feature of the QoE dashboard is the ability to quickly identify reductions or changes in application performance (before users start calling), and determine if the change is caused by an increase in network delay or slow Application Server performance. This answers the question: Is it the network or the application? You can also easily view how current conditions compare to historical trends by checking four Top 10 graphs.

The system provides Top 10 Application Response Time, Network Response Time, Data Volume, and Transactions graphs. There are also tables that support analysis and comparison of Average and Peak network, application response times, and various loadings —Average and Total Data Volumes, Average Transactions/Min, and Total # of Transactions—to other performance metrics.

The graphs feature color-coded lines for each listed application showing 5-minute resolution sample value trends. Hovering over a point on a line opens an information window that shows the Date-Timestamp, application name, and the data value for that sample. A sliding control under the graph allows range adjustment of the viewable timeframe.



QOE APPLICATION	AVERAGE APPLICATION RESPONSE TIME	PEAK VALUE
🟦 ☑ CIFS	2.27 s	7.90 s
🟦 ☑ SW NPM	N/A	N/A
🟫 ☑ FTP	N/A	N/A
🟨 ☑ MS SQL	N/A	N/A
⬛ ☑ HTTP	N/A	N/A

Top 10 Average Application Response Time Table

The table under each graph lists the applications included in the Top 10 synopsis, along with Average and Peak values. Average values that exceed a set performance threshold are highlighted.

The QoE Dashboard includes a table that lists all of the monitored Nodes, a table of Nodes Exceeding Thresholds, and a Quality of Experience Application Stats table that summarizes the four major performance metrics: Network Response Times, Application Response Times, Total Data Volumes, and Total # of Transactions by Application.

Traffic Categories

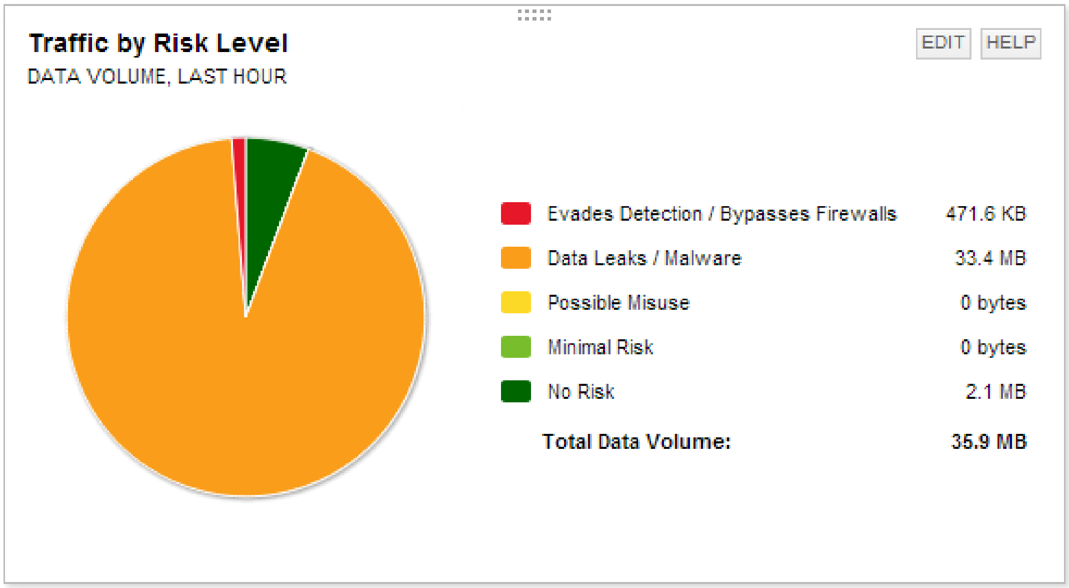
Each QoE-monitored application—whether selected from the very large list of pre-configured applications, or a custom user-configured HTTP application—is identified as a specific type of each of three categories:

Category: Collaboration, Database, File Transfer, Games, Mail, Messaging, Network Monitoring, Networking, Proxy, Remote Access, Social Networking, Streaming Media, VPN, and Web Services.

Risk Level: No Risk, Minimal Risk, Possible Misuse, Data Leaks/Malware, Evades Detection/Bypasses Firewalls.

Productivity Rating: All Social, Mostly Social, Both Business and Social, Mostly Business, All Business.

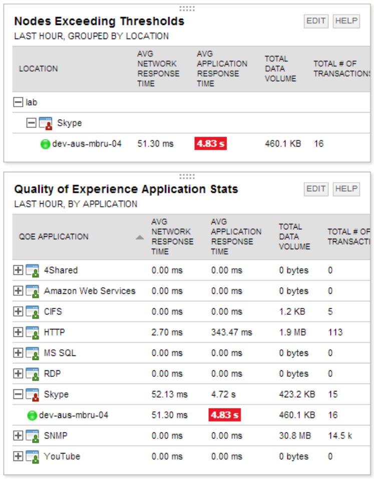
The QoE Console also includes three pie charts and tables to allow quick identification of relative and specific volumes Traffic. Traffic is categorized as Business Related vs Social Traffic, Traffic by Category, and Traffic by Risk Level—the latter being especially helpful for quickly identifying uncharacteristic increases in potentially malicious traffic.



Traffic by Risk Level
DATA VOLUME, LAST HOUR

🟥 Evades Detection / Bypasses Firewalls	471.6 KB
🟧 Data Leaks / Malware	33.4 MB
🟨 Possible Misuse	0 bytes
🟩 Minimal Risk	0 bytes
🟩 No Risk	2.1 MB
Total Data Volume:	35.9 MB

Traffic Volume by Risk Level



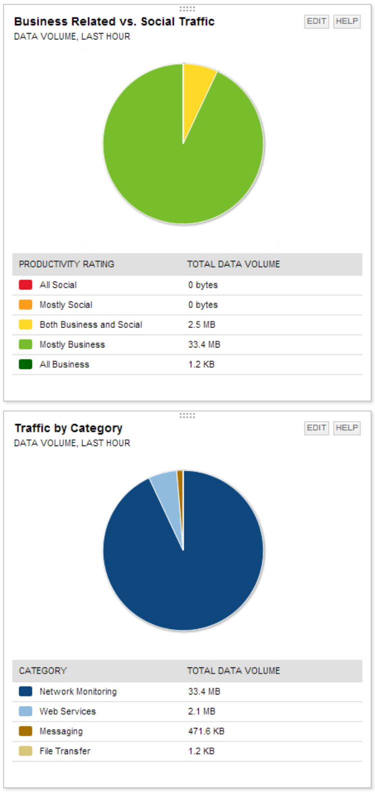
Nodes Exceeding Thresholds
LAST HOUR, GROUPED BY LOCATION

LOCATION	AVG NETWORK RESPONSE TIME	AVG APPLICATION RESPONSE TIME	TOTAL DATA VOLUME	TOTAL # OF TRANSACTIONS
⊟ lab				
⊟ Skype				
● dev-aus-mbru-04	51.30 ms	4.83 s	460.1 KB	16

Quality of Experience Application Stats
LAST HOUR, BY APPLICATION

QOE APPLICATION	AVG NETWORK RESPONSE TIME	AVG APPLICATION RESPONSE TIME	TOTAL DATA VOLUME	TOTAL # OF TRANSACTI
⊞ 4Shared	0.00 ms	0.00 ms	0 bytes	0
⊞ Amazon Web Services	0.00 ms	0.00 ms	0 bytes	0
⊞ CIFS	0.00 ms	0.00 ms	1.2 KB	5
⊞ HTTP	2.70 ms	343.47 ms	1.9 MB	113
⊞ MS SQL	0.00 ms	0.00 ms	0 bytes	0
⊞ RDP	0.00 ms	0.00 ms	0 bytes	0
⊟ Skype	52.13 ms	4.72 s	423.2 KB	15
● dev-aus-mbru-04	51.30 ms	4.83 s	460.1 KB	16
⊞ SNMP	0.00 ms	0.00 ms	30.8 MB	14.5 k
⊞ YouTube	0.00 ms	0.00 ms	0 bytes	0

QoE Thresholds and Application Stats tables



Business Related vs. Social Traffic
DATA VOLUME, LAST HOUR

PRODUCTIVITY RATING	TOTAL DATA VOLUME
🟥 All Social	0 bytes
🟧 Mostly Social	0 bytes
🟨 Both Business and Social	2.5 MB
🟩 Mostly Business	33.4 MB
🟩 All Business	1.2 KB

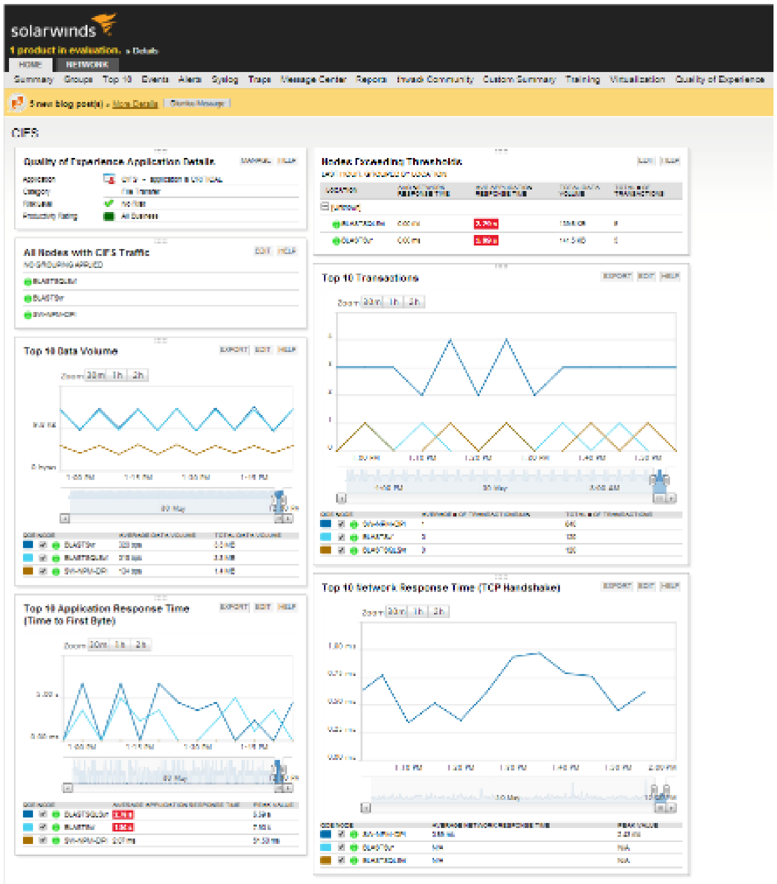
Traffic by Category
DATA VOLUME, LAST HOUR

CATEGORY	TOTAL DATA VOLUME
🟦 Network Monitoring	33.4 MB
🟦 Web Services	2.1 MB
🟫 Messaging	471.6 KB
🟨 File Transfer	1.2 KB

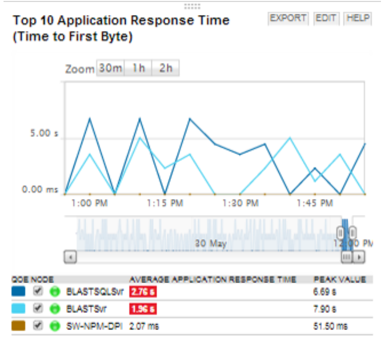
Traffic Volume by Productivity Rating and Category charts

Application Dashboard

Clicking on an application name in any of the Top 10 tables (CIFS, for example) on the primary QoE dashboard opens an application-specific dashboard that provides the same four Top 10 graphs and tables (Data Volume, Transactions, Application Response Times, and Network Response Times) for that application. It also includes Application Details (Name, Category, Risk Level, and Productivity Rating), a list of the Nodes providing the target application, and a summary table of any Nodes which are exceeding pre-set thresholds for each of the four performance metrics.


solarwinds
1 product in evaluation. » Details
HOME
NETWORK
Summary
Groups
Top 10
Events
Alerts
Syslog
Traps
Message Center
Reports
thwack Community
Custom Summary
Training
Virtualization
Quality of Experience
5 new blog post(s) » More Details
CIFS
Quality of Experience Application Details
File Transfer
All Business
Nodes Exceeding Thresholds
All Nodes with CIFS Traffic
Top 10 Transactions
Top 10 Data Volume
Top 10 Network Response Time (TCP Handshake)
Top 10 Application Response Time (Time to First Byte)
EXPORT
EDIT
HELP
Zoom 30m 1h 2h
30 May

Hovering over the lines in any of these graphs opens an information window that depicts the specific nodes and their respective values for each metric.



QOE NODE				AVERAGE APPLICATION RESPONSE TIME	PEAK VALUE
🟦	☑	●	BLASTSQLSvr	2.76 s	6.69 s
🟦	☑	●	BLASTSvr	1.96 s	7.90 s
🟫	☑	●	SW-NPM-DPI	2.07 ms	51.50 ms

Top 10 CIFS Application Response Times Nodes

Share:

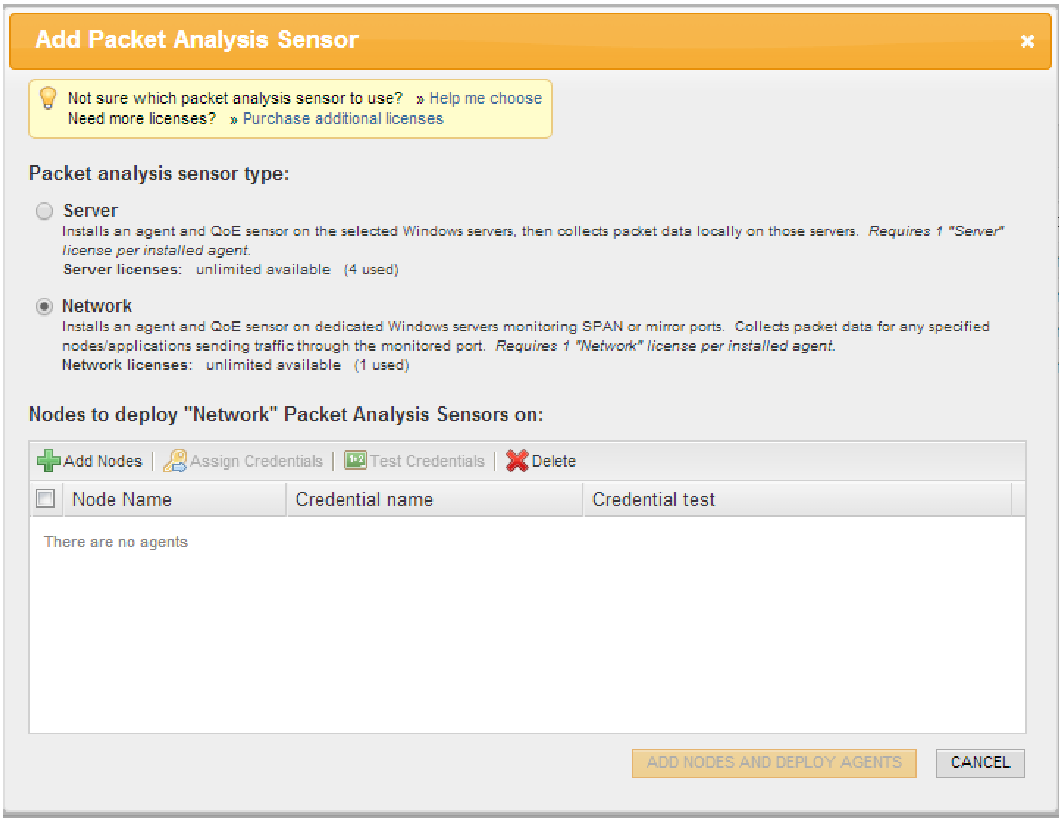
Setting up QoE

Setting up a Quality of Experience environment is as simple as using a step-by-step wizard to deploy sensors, and selecting pre-configured or customized settings to monitor applications.

Deploying Packet Analysis Sensors

There are two deployment options:

- **Packet Analysis Sensors for Networks**—used to monitor packet traffic using a SPAN/mirror port and a dedicated sensor installed on a Windows server
- **Packet Analysis Sensors for Servers**—used to monitor packet traffic directly on your Windows server or workstation



Add Packet Analysis Sensor ×

Not sure which packet analysis sensor to use? » Help me choose
Need more licenses? » Purchase additional licenses

Packet analysis sensor type:

○ Server
Installs an agent and QoE sensor on the selected Windows servers, then collects packet data locally on those servers. *Requires 1 "Server" license per installed agent.*
Server licenses: unlimited available (4 used)

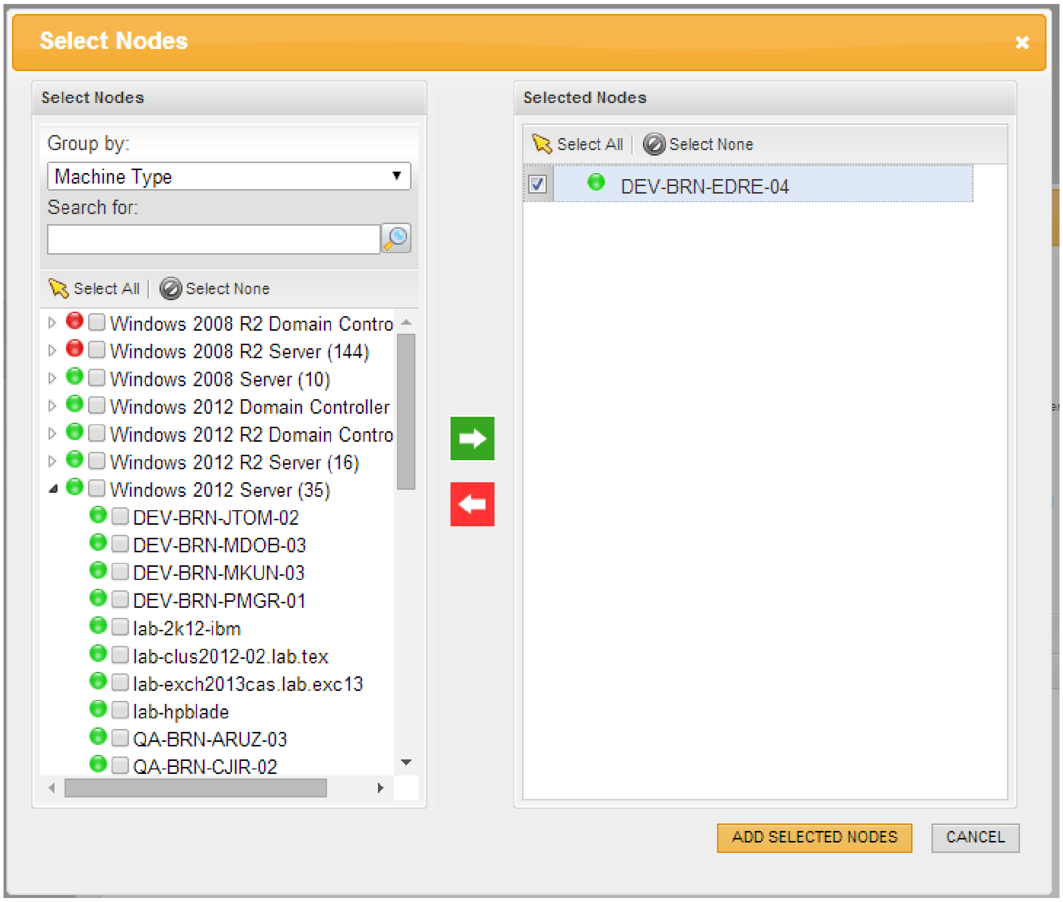
● Network
Installs an agent and QoE sensor on dedicated Windows servers monitoring SPAN or mirror ports. Collects packet data for any specified nodes/applications sending traffic through the monitored port. *Requires 1 "Network" license per installed agent.*
Network licenses: unlimited available (1 used)

Nodes to deploy "Network" Packet Analysis Sensors on:

□	Node Name	Credential name	Credential test

There are no agents

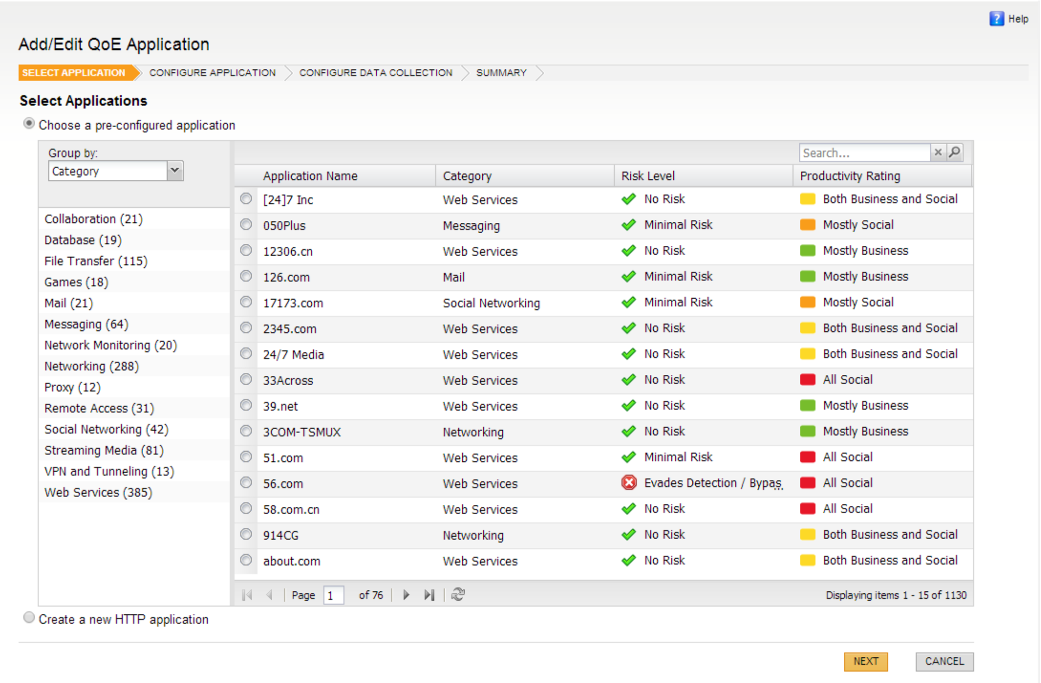
To begin installing a network packet analysis sensor, simply select Add Node, select the node to install the agent, and add the selected node. You'll then be asked to provide the node credentials. After doing so, you can then deploy the agent.



Manage QoE Applications

The next step is to select the applications from which you are going to analyze traffic.

You can choose a pre-configured application on the Manage QoE Applications screen which allows for a selection from a very large list of applications available out of the box. A Search box that supports partial strings makes finding the right application quick and easy.



Add/Edit QoE Application

SELECT APPLICATION > CONFIGURE APPLICATION > CONFIGURE DATA COLLECTION > SUMMARY

Select Applications

● Choose a pre-configured application

Group by: Category

- Collaboration (21)
- Database (19)
- File Transfer (115)
- Games (18)
- Mail (21)
- Messaging (64)
- Network Monitoring (20)
- Networking (288)
- Proxy (12)
- Remote Access (31)
- Social Networking (42)
- Streaming Media (81)
- VPN and Tunneling (13)
- Web Services (385)

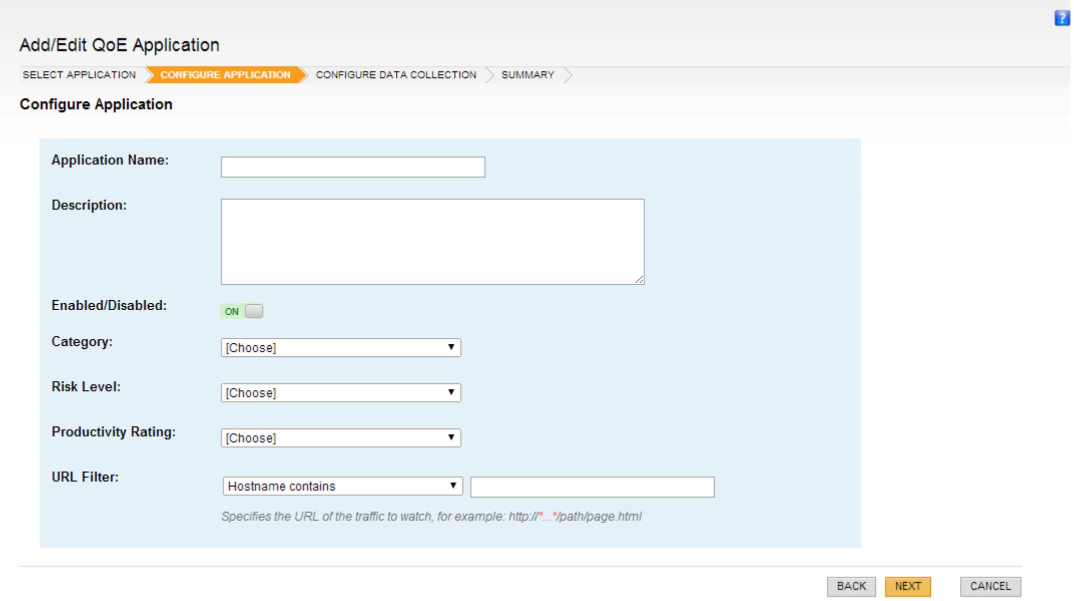
Search...

	Application Name	Category	Risk Level	Productivity Rating
○	[24]7 Inc	Web Services	✓ No Risk	Both Business and Social
○	050Plus	Messaging	✓ Minimal Risk	Mostly Social
○	12306.cn	Web Services	✓ No Risk	Mostly Business
○	126.com	Mail	✓ Minimal Risk	Mostly Business
○	17173.com	Social Networking	✓ Minimal Risk	Mostly Social
○	2345.com	Web Services	✓ No Risk	Both Business and Social
○	24/7 Media	Web Services	✓ No Risk	Both Business and Social
○	33Across	Web Services	✓ No Risk	All Social
○	39.net	Web Services	✓ No Risk	Mostly Business
○	3COM-TSMUX	Networking	✓ No Risk	Mostly Business
○	51.com	Web Services	✓ Minimal Risk	All Social
○	56.com	Web Services	☒ Evades Detection / Bypas	All Social
○	58.com.cn	Web Services	✓ No Risk	All Social
○	914CG	Networking	✓ No Risk	Both Business and Social
○	about.com	Web Services	✓ No Risk	Both Business and Social

Page 1 of 76 Displaying items 1 - 15 of 1130

○ Create a new HTTP application

Clicking 'Create a new HTTP application' in this screen allows the creation of a custom QoE-monitored application. This is done by entering the application name, description, selecting a few settings from the three available categories to properly classify the application for reporting purposes, and selecting a qualifier and URL Filter string to tell the QoE system how to collect the desired performance metrics for the application.



Add/Edit QoE Application

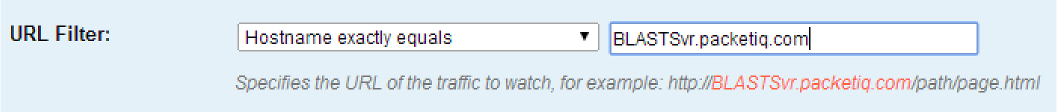
SELECT APPLICATION > CONFIGURE APPLICATION > CONFIGURE DATA COLLECTION > SUMMARY

Configure Application

Application Name:	
Description:	
Enabled/Disabled:	ON
Category:	[Choose]
Risk Level:	[Choose]
Productivity Rating:	[Choose]
URL Filter:	Hostname contains

Specifies the URL of the traffic to watch, for example: http://...*/path/page.html*

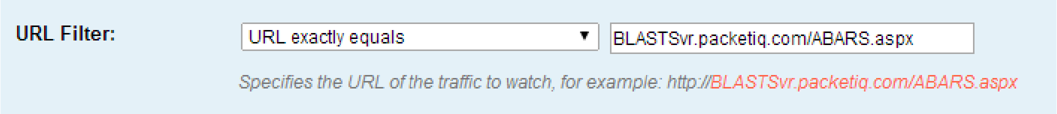
The URL Filter setting allows you to configure the QoE Agent to identify your application by either a Hostname or URL string.



URL Filter: Hostname exactly equals ▼ BLASTSvr.packetiq.com

Specifies the URL of the traffic to watch, for example: http://BLASTSvr.packetiq.com/path/page.html

Using the URL option allows specifying a sub-set or a particular path and page out of all the possible paths/pages that may be hosted on that URL.



URL Filter: URL exactly equals ▼ BLASTSvr.packetiq.com/ABARS.aspx

Specifies the URL of the traffic to watch, for example: http://BLASTSvr.packetiq.com/ABARS.aspx

After configuring the application, one or more Nodes that should collect this application traffic can be selected. Clicking ‘Finish’ completes the configuration and starts the collection process.

Summary

The SolarWinds Quality of Experience solution adds complex packet-level analysis capabilities to Network Performance Monitor (NPM). This allows the support of integrated network and application performance monitoring across the enterprise.

Having the QoE dashboard integrated into the NPM solution allows for both automated identification of performance issues and drill-down capability. This allows the identification of the true source and nature of the performance issue so that problems can be quickly addressed—before users start calling.

Best of all, the SolarWinds QoE solution offers the advantages of sophisticated deep-packet analysis to a segment of the market that historically has not been able to afford packet inspection solutions. This helps level the playing field in markets where application services performance and reliability is a critical competitive factor.

SolarWinds Network Performance Monitor

Get a free, fully functional 30-day trial of SolarWinds Network Performance Monitor.

- Simplifies detection, diagnosis, & resolution of network issues before outages occur
- Tracks response time, availability, & uptime of routers, switches, & other SNMP-enabled devices
- Shows performance statistics in real time via dynamic, drillable network maps
- Includes out-of-the-box dashboards, alerts, reports, & expert guidance on what to monitor & how
- Automatically discovers SNMP-enabled network devices & typically deploys in less than an hour

Learn More

About SolarWinds

SolarWinds (NYSE: SWI) provides powerful and affordable IT management software to customers worldwide from Fortune 500 enterprises to small businesses. In all of our market areas, our approach is consistent. We focus exclusively on IT Pros and strive to eliminate the complexity that they have been forced to accept from traditional enterprise software vendors. SolarWinds delivers on this commitment with *unexpected simplicity* through products that are easy to find, buy, use and maintain while providing the power to address any IT management problem on any scale. Our solutions are rooted in our deep connection to our user base, which interacts in our online community, **thwack**®, to solve problems, share technology and best practices, and directly participate in our product development process. Learn more today at **http://www.solarwinds.com/.**

Learn More

For product information or to purchase SolarWinds products, visit solarwinds.com, or reach out to us at:

Americas

Phone: 866.530.8100

Fax: 512.682.9301

Email: **sales@solarwinds.com**

EMEA

Phone: +353 21 5002900

Fax: +353 212 380 232

Email: **sales@solarwinds.com**

APAC

Tel : +65 6593 7600

Fax : +65 6593 7601

Email: **sales@solarwinds.com**

Federal, Federal Reseller
and System Integrator Sales

Phone: 877.946.3751

Fax: +353 212 380 232

Email: **federalsales@solarwinds.com**

7171 Southwest Parkway, Building 400, Austin, Texas 78735