



Product Manual

Version 5

This documentation is a work in progress.
If you have any questions that are not currently addressed here, please don't
hesitate to contact us at support@pingman.com.

PingPlotter

(c) 1998, 2016 Pingman Tools LLC

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: May 2016

Publisher

Pingman Tools, LLC

Managing Editor

Pete Ness

Technical Editor

Gary Welch

Special thanks to:

All the people who contributed to this documentation and to the PingPlotter software.

Special recognition to:

Pierre Schwab

Tom Larrow

Jenna

Clara

Conrad

Andrea, D.D., Nina

Mele

Cam M.

Larry M.

Table of Contents

Part I Introduction	2
Part II 90 Second Overview	5
Part III How PingPlotter Works	12
Part IV Operation	16
1 Downloading and Installing	16
2 Target Bar	17
3 Trace Windows	18
4 Time Line Graphs	20
5 Summary Graphs	23
6 Workspaces	25
7 Web Interface	27
Part V Common Tasks	31
1 Alerts	31
Creating / Configuring Alerts	31
Email Setup (for alerts)	39
Event Notification	41
Troubleshooting Alerts	42
Alert Events	43
Play a Sound.....	43
Send Email.....	43
Log to file	44
Tray Icon Change.....	44
Web REST Call.....	45
Add/Remove from Summary.....	46
2 Variable Substitution	47
3 Auto-Saving of Data	48
4 Running as a service	52
5 Handling Multiple Targets	54
6 Named Configurations	55
7 Reporting	57
8 Exporting data for further analysis	58
9 Route Changes	60
Part VI Understanding Output	63
1 Discovering a route between you and a target	63
2 Interpreting Results - A Quick Example	64

3	Finding the source of the problem	66
4	Finding the source of the problem - part 2	69
5	Interpreting Results - Longterm Monitoring	71
6	Interpreting Results - Gamers	73
7	Interpreting Results - ISP Problems	76
8	Interpreting Results - Bad Hardware	81
9	Interpreting Results - Bandwidth Saturation	86
10	Voice over IP (VoIP) troubleshooting	89
11	Building a Compelling Case	101

Part VII Advanced Features 104

1	IPv6 and IPv4	104
2	Raw TCP sockets and WinPcap	104
3	Tracing from a remote server	105
4	Command Line Arguments	113
5	Installer Options and MSI	114

Part VIII Reference 117

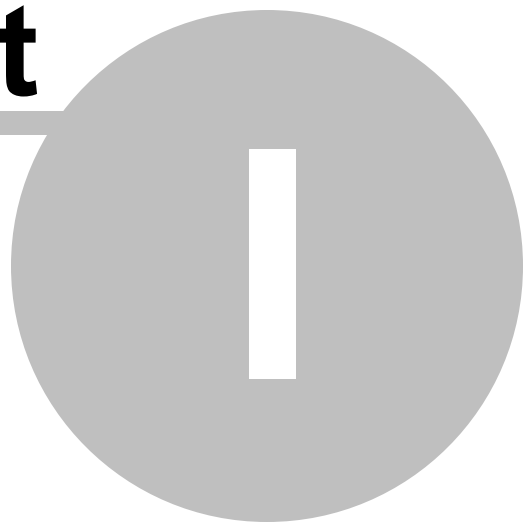
1	Options	117
	General Options	118
	Web Server Options	119
	Email Settings	120
	Display Options	122
	Engine Options	125
	Auto-Save Options	128
2	Menus	129
	File Menu	129
	Edit Menu	130
	Workspace Menu	131

Part IX More Information 133

1	Support	133
2	Purchasing	133

Index 135

Part



Introduction

1 Introduction

A network connection probably isn't something you think about on a regular basis. When it isn't working right (or stops working all-together)... well, that's a completely different story. At that point, it's **all** you can think about. The only thing that's worse than your network connection not working? Trying to figure out why it's **not** working.

If you're reading this, odds are you've encountered (or are currently encountering) this exact situation. Or maybe you're just preparing yourself ahead of time (go you!). No matter your current network situation, PingPlotter can help you get to the bottom of these kinds of problems faster, so you can find a solution and get back to not having to think about your connection.

PingPlotter was originally created in 1998 to troubleshoot unacceptable lag in an online game (a problem which that particular ISP was claiming no responsibility for at the time). Over the years, the program has grown, and added a variety of features and capabilities. Today, it's a very powerful network monitoring, troubleshooting, and diagnostic tool, which is used by a variety of users - from the "weekend troubleshooter" to full-time network administrators.

PingPlotter can help with a variety of different network related woes - and can be a great help to you if:

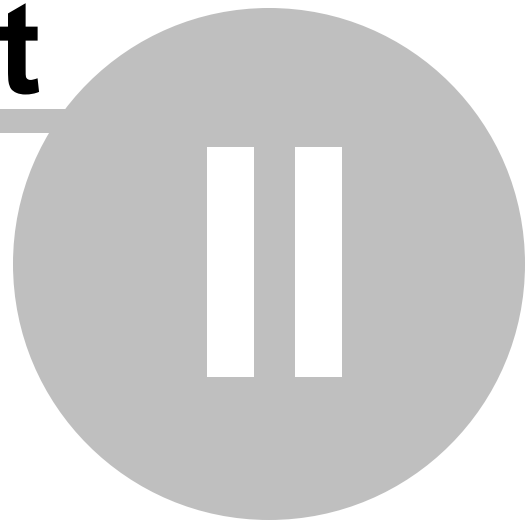
- You rely on a network or internet service, which happens to be having problems - such as slow performance, random disconnects, or other similar issues.
- You're a systems administrator - and you need to know when connectivity to one of your servers go down (and want some evidence of where/when/why it went down).
- A provider is telling you that they can't see any problems (when you're clearly having issues) - and you need to show them where the problem really is.
- In general, if you're a user of something that relies heavily on a network or the internet, such as:
 - A web browser
 - VoIP services/video chat
 - Online gaming
 - Streaming audio/video
 - An ASP for your business (such as payroll, accounting, human resources, etc)
 - Home automation products

If you're here, one of the above bullet points most likely applies to you. There are, however, a few

scenarios where PingPlotter may **not** be the right tool; if you have hundreds (or thousands) of network nodes with many services you need to monitor, if you need true SNMP capability (although [PingPlotter can trigger SNMP traps via alerts](#)), or if you need auto-discovery of network nodes.

If you ever have any questions or comments concerning PingPlotter, this guide, or if you just want to email someone to say "hello" - please feel free to send us an email at info@pingplotter.com. We're always happy to answer any questions or provide any advice that we can!

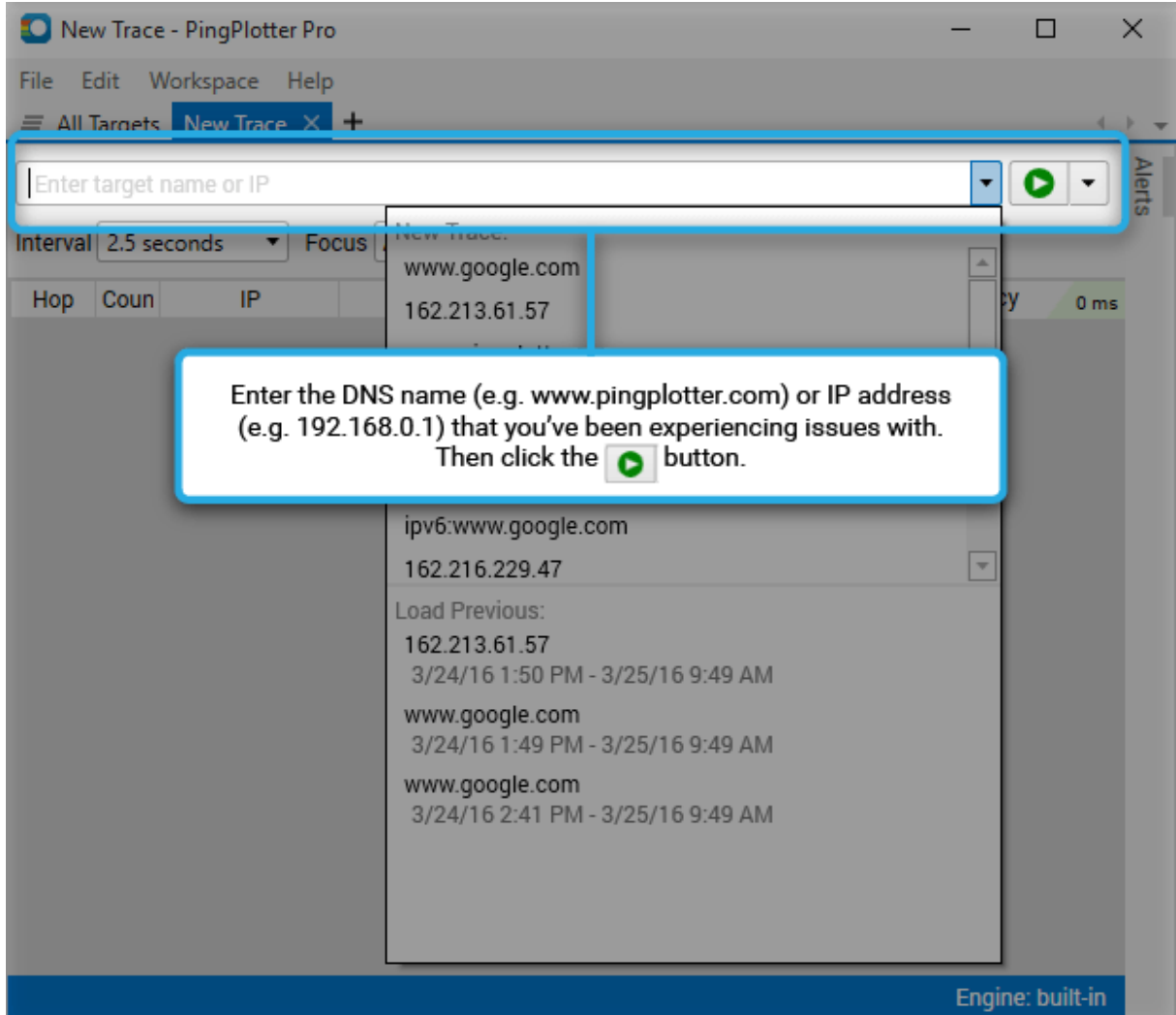
Part



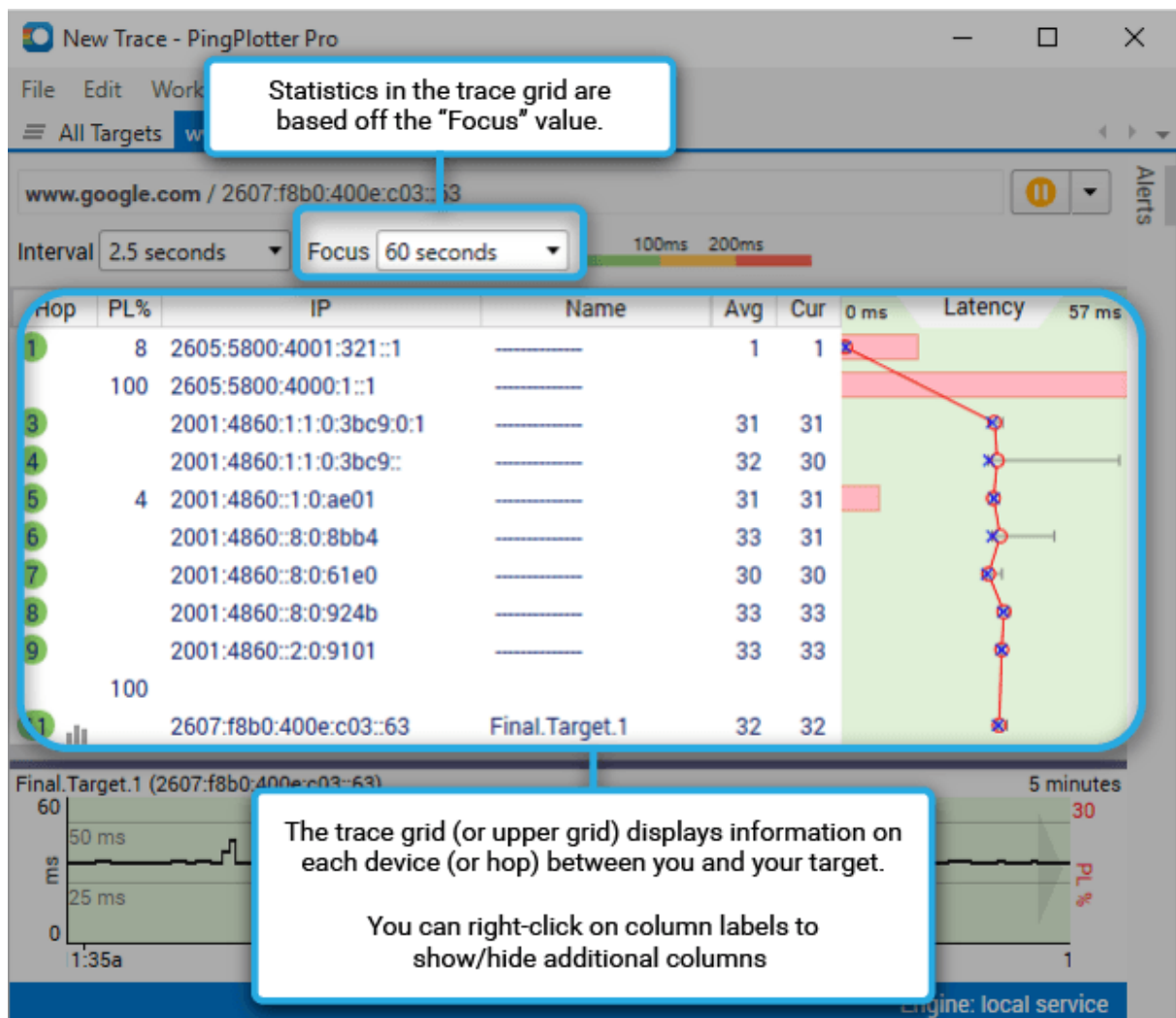
90 Second Overview

2 A 90 Second Overview of PingPlotter

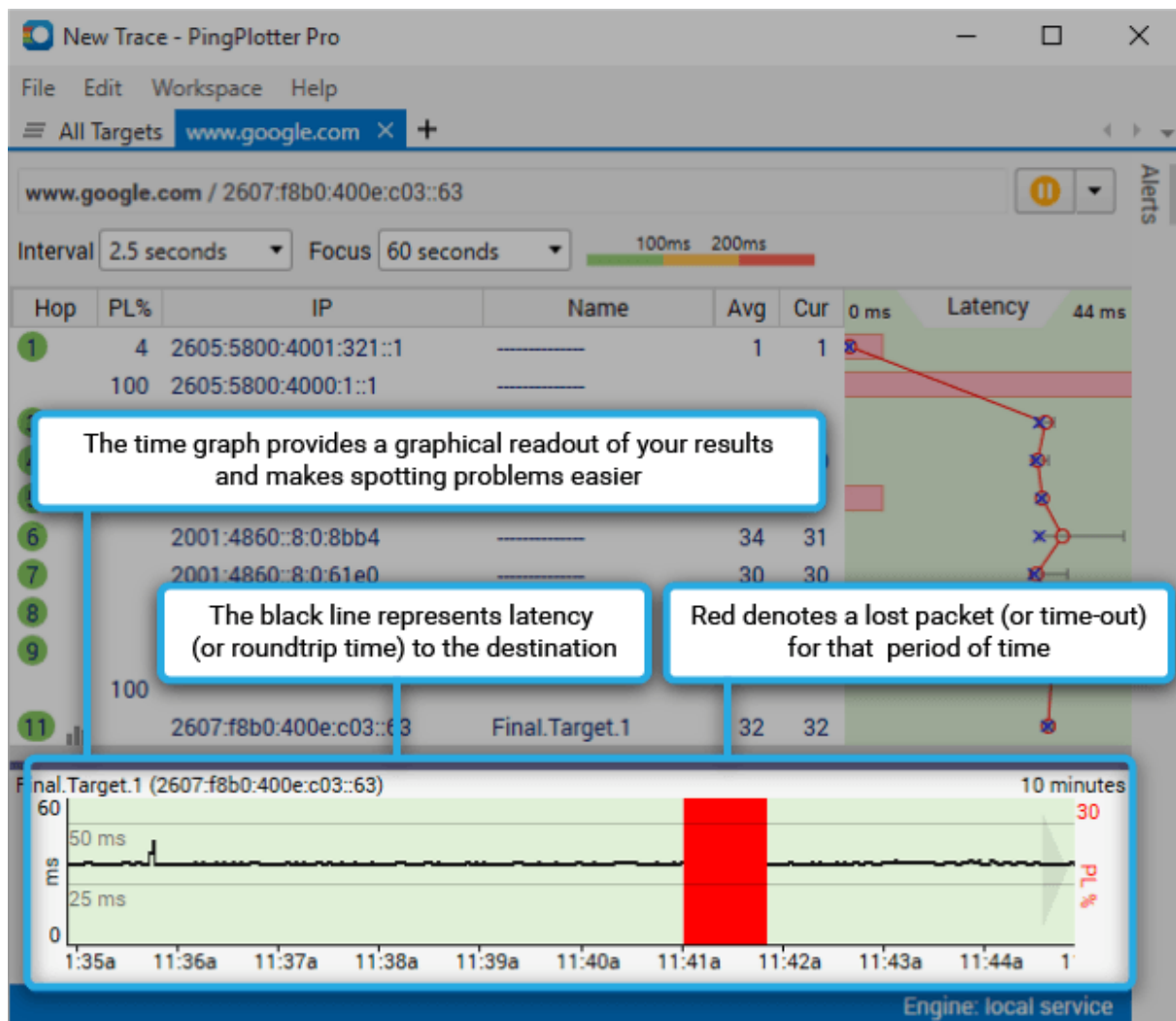
Once you've got PingPlotter [downloaded and installed](#) , starting to collect data is a breeze.

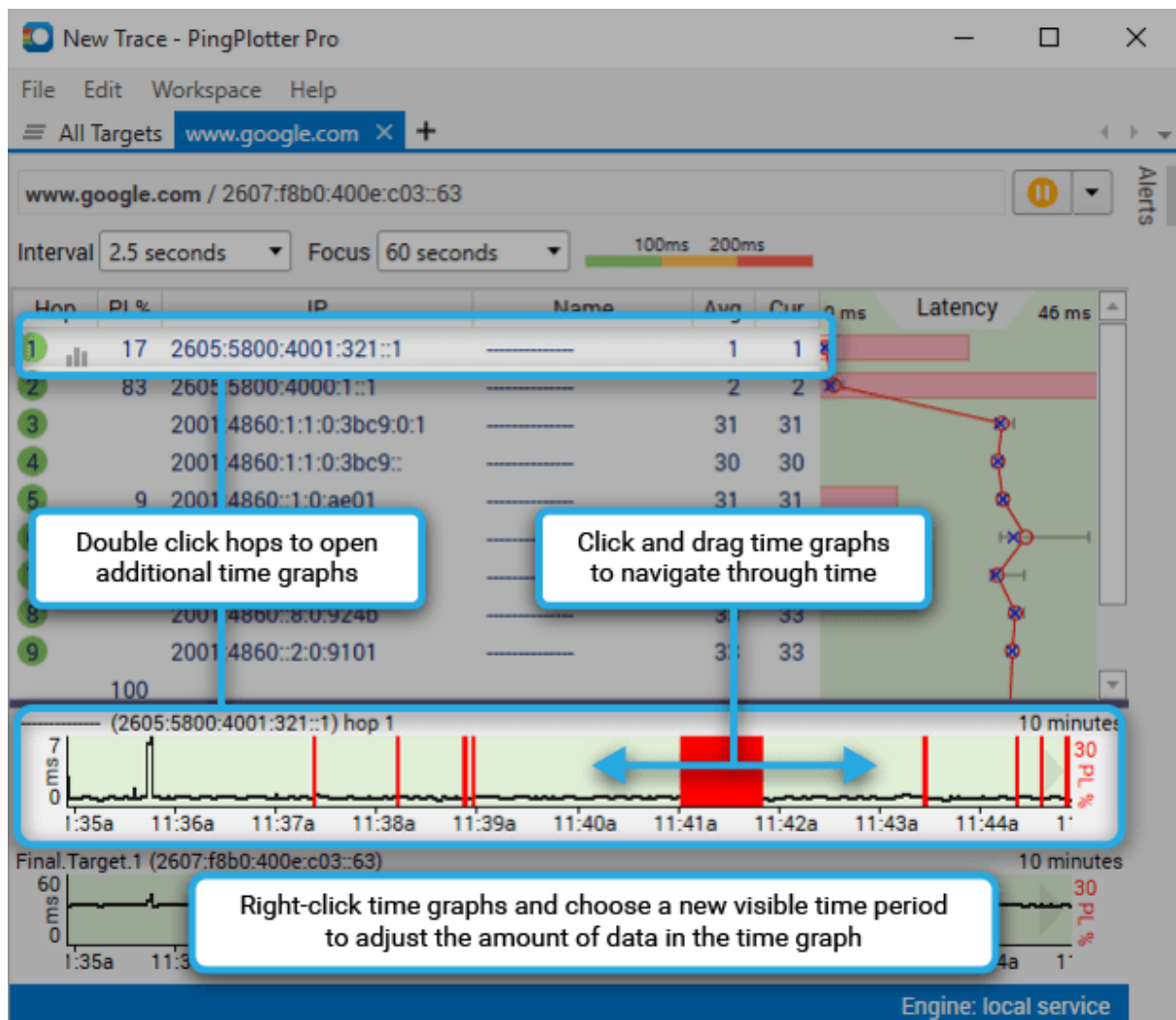


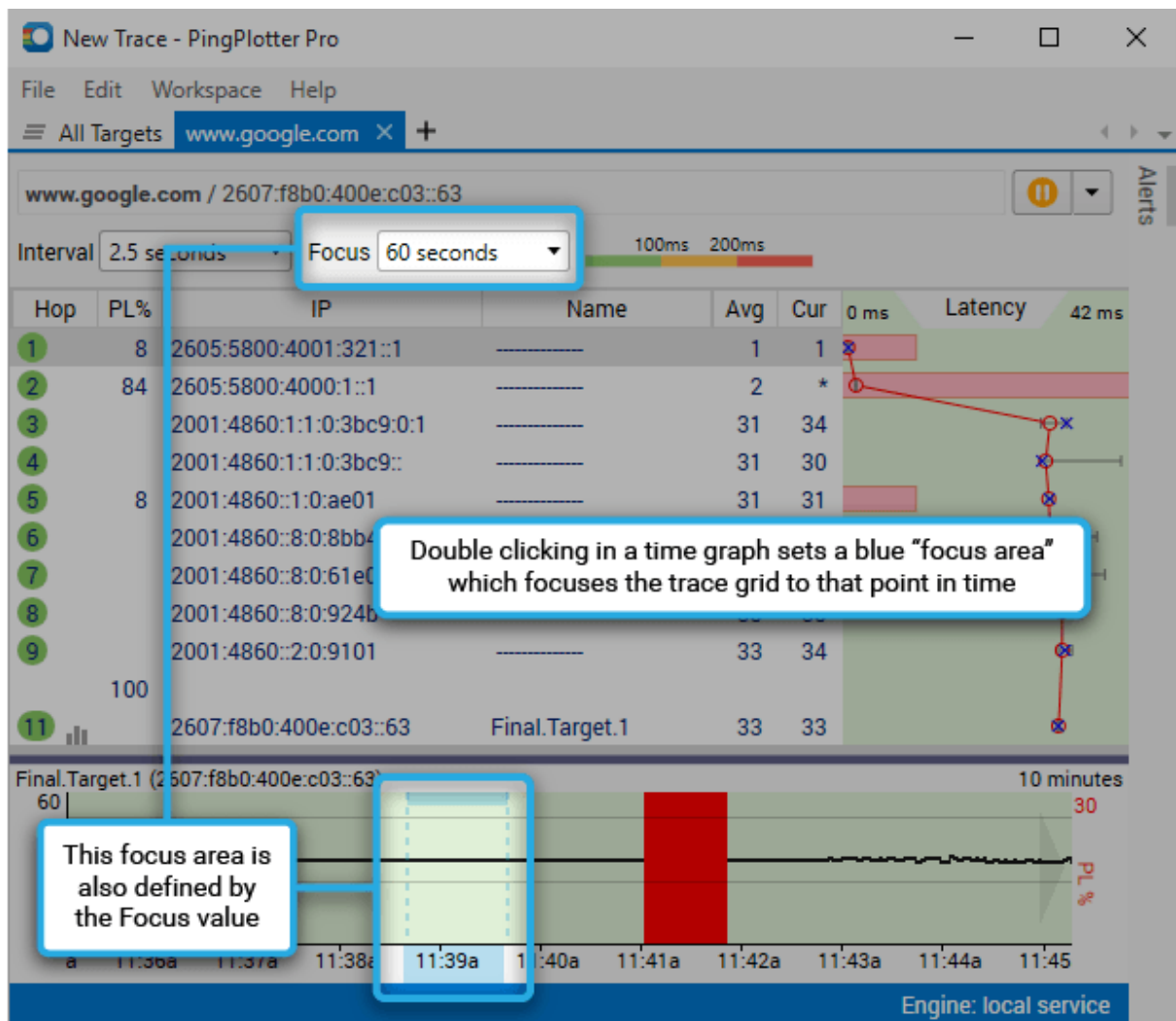
PingPlotter will then begin to generate route information. If you're getting a "Destination address unreachable" message - [have a look here](#) for some possible solutions.

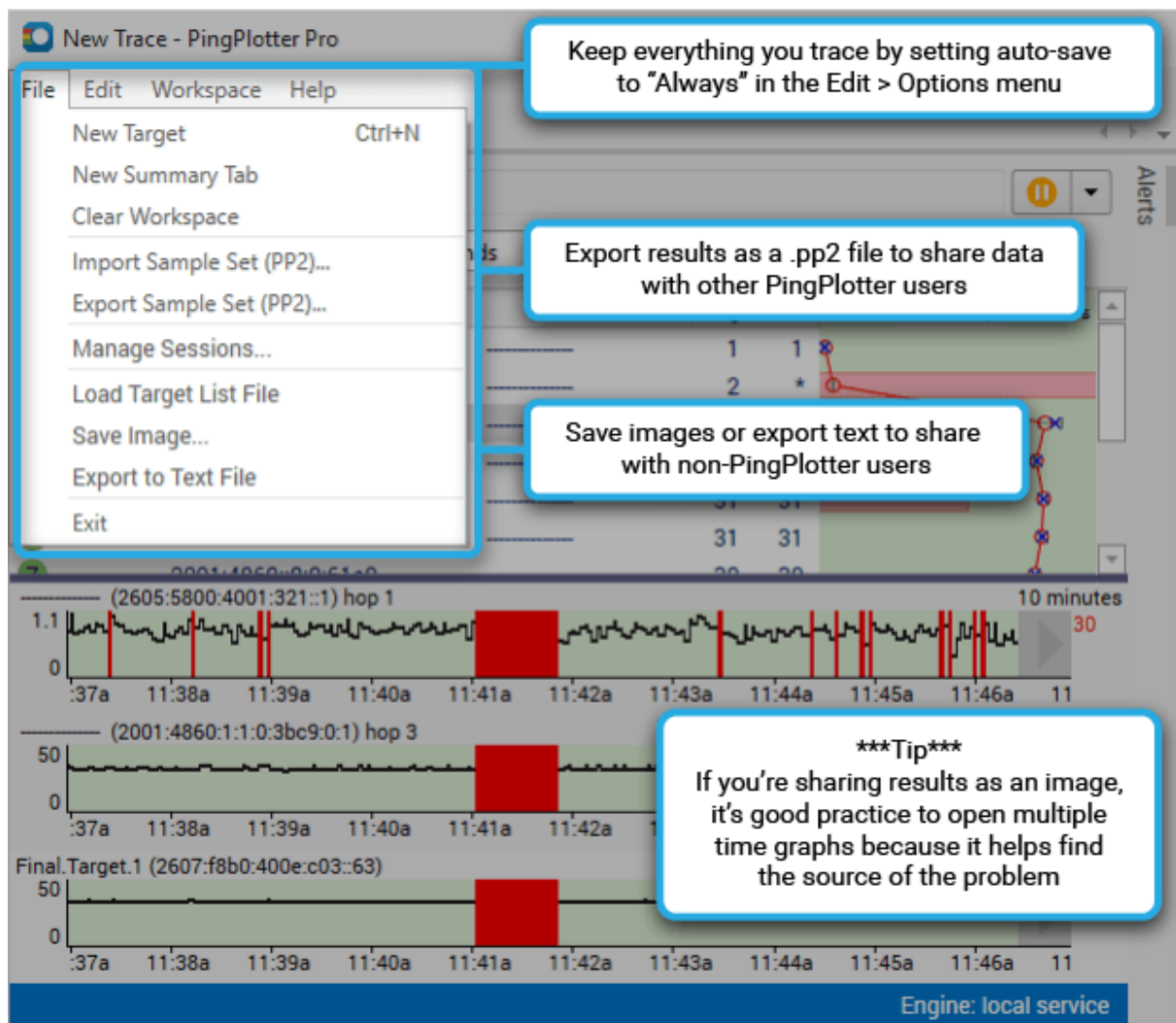


Worried about the packet loss that you see at hop #4 and #10 in the above screen shot? As long as it doesn't seem to be affecting our final destination (which it isn't here) - then it's nothing to worry about. See this [Knowledge Base article](#) for more information.



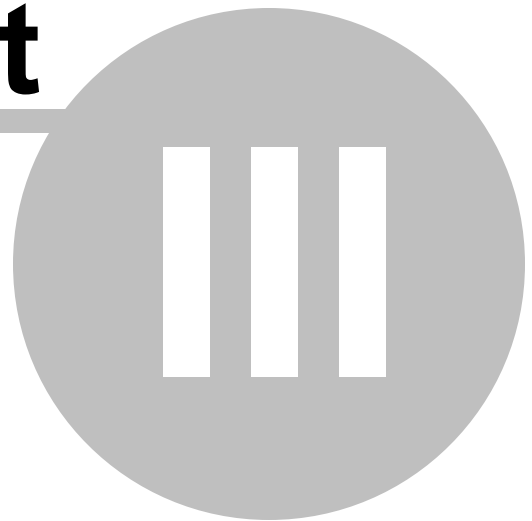






****Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details****

Part



How PingPlotter Works

3 How PingPlotter Works

If you're relatively new to the world of troubleshooting network problems, a lot of this subject matter can seem a bit daunting at first. Fret not, though - we've got an analogy that should help make things "click."

If you're not new to this space (and possibly just rolled your eyes at the idea of reading though an analogy), feel free to skip to the technical details toward the bottom of this section.

Network traffic is a bit like freeway traffic

A network can operate a bit like a freeway; things are great when everyone is going the speed limit and we only have 50% of the maximum traffic that's designed to go on that freeway. When we start to add more and more traffic, though, at some point the freeway won't have the capacity for any more.

Problems will start to arise as new drivers try to merge on the freeway. People that are already on the freeway will sometimes slow down and cause traffic jams. If it gets too bad, some people may give up all together - and decide not to continue their journey.

On a freeway, this might be referred to as "congestion" - and this "congestion" happens on networks as well (in pretty much the same way). Packet loss and latency (two terms you'll be getting very familiar with) are both symptoms of congestion - when there's too much traffic for the network to handle.

Lets say this proverbial freeway is the one we take to get home from work everyday. Let's also say (for this example) that between work and home there are 15 off-ramps with turnaround points off of the freeway. Lets also say that we've got a team of 15 people, each with their own car, ready to do whatever we ask of them (again, for the sake of this example. Stick with us here).

If we want to find out the conditions on the freeway for our drive home from work, we could send out 15 cars, and assign each driver to one of those 15 off-ramps. The instructions for each driver are the same: get you your assigned off-ramp, turn around, and come back. Then we'll measure the time it takes each car to get from us, to their off-ramp, and then back to us.

The most important car is the one that goes all the way to your target (or home) - that 15th car. If it makes it there and back again in the expected time, then we know that the traffic on the freeway is running pretty well.

If that 15th car takes longer than expected (or if it never returns), then we can look at the results for the other off-ramps to find a likely place where problems could be occurring. Maybe all the cars through off-ramp #9 had no problems (and returned quickly) - but the cars that went through off-ramp #10 (and beyond) started getting delays (or didn't return). From this, we can see that there is some kind of

problem happening past off-ramp #9.

PingPlotter operates *very* similar to this; it sends out data packets that go all the way to a target destination (as well as each stop in between), and measures the amount of time it takes to get there. It also measures how often a packet (or a router) gives up. This information can be used to figure out where (and when) problems are occurring.

For the sake of taking this analogy too far, let's look at one other scenario. Let's say that off-ramp #5 is in a small town where the police are of a disposition to pull people over for no reason (this is all theoretical, of course - we would never imply that any police actually do this). Each car that we send to off-ramp #5 have to pass by these police officers, and 20% of the time, they get pulled over. Another 15% of the time, there's someone else pulled over there, and our car has to wait while that car moves off the road. Meanwhile, traffic is whizzing by on the freeway, unrestricted.

This situation can happen on a network with PingPlotter as well - where the packets going to hop 5 might get waylaid by some local rules and show packet loss, latency, and jitter that are not being experienced by packets destined for other places.

Technically speaking...

At its heart, PingPlotter is a trace route utility. It's souped up and on steroids, but the basic data it collects is based on the theory of trace route.

A ping packet is an IP packet requesting that a copy of its contents be echoed back to the sender. When you "ping" a site, you send over an echo request and that site responds back that it received it.

One of the parameters on a ping packet is something called "Time to live" (TTL) - which is an IP header field designed to keep packets from running in loops (essentially forever) throughout a network (this can happen when there is a route change, and the routers involved don't all know the same information as new information is being replicated out). Initially it's usually set to somewhere between 64 and 255, and is reduced by 1 every time it passes through a server. If the TTL should ever reach zero, the packet has expired, and the router that it's passing through will send it back to the source.

Trace route plays with this TTL number on outgoing packets. It first sends out a packet with a TTL of 1. The first router that sees this and decrements it to 0, and then sends it back. It also sends back its own IP address with the packet, and DNS is used to do a lookup for an actual domain name.

Next, traceroute sends out a packet with a TTL of 2 so it can find out what the next computer in the route is. Then it sends out a packet with a TTL of 3. This process is repeated until the final destination is reached. At that point, you know the entire path the packet has traversed to reach the destination

computer/router. Each server/router in this chain is called a hop.

This method can help us determine the route a packet takes, but if we time each of these packets, we also know how long it takes for a packet to make it from our source PC, to that router, and then back again. This is called latency.

The last hop in a (successful) trace route is actually the round-trip time to the destination server. This is an important concept to understand. You don't add up all the times between you and the destination host - as that time has already been added. The time to the last hop in the chain is exactly the same as is if you'd used a ping utility to that host. So a trace route utility is actually two utilities - ping AND trace route.

PingPlotter speeds up this process by sending out packets to the first 35 servers in the route all at the same time. This makes a HUGE difference in overall speed. It also means that the network conditions for each hop are very similar - so the numbers are better compared.

Part

IV

Operation

4 Operation

4.1 Downloading and Installing

The latest version of PingPlotter is always accessible via our [download page](#). If you're a licensed user and need to download an older version of the program - these versions can be found by accessing your [account page](#).

Once downloaded and launched, the PingPlotter installer starts a wizard driven install that steps through the installation process. We suggest leaving all the settings at their defaults (if you want to install PingPlotter as a Windows service, [have a look here](#)).

At the end of the install, you'll be asked if you want to "Run PingPlotter now". If for some reason you don't want to at that time, uncheck that checkbox. Otherwise congratulations, you now should have a PingPlotter group in with the rest of your installed applications (i.e.: Start menu, Programs group, PingPlotter folder) and are ready to go!

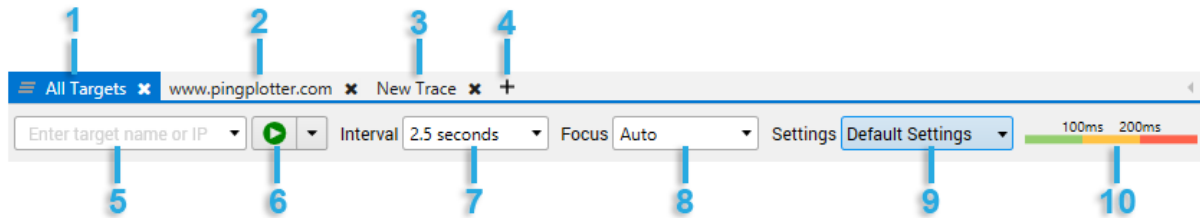
If this procedure doesn't work for you for some reason (if you get a weird error, etc.), please see our knowledge base article [here](#). If you still encounter problems with the download, please send an email to support@pingplotter.com describing the specific error you're getting, and/or any symptoms that you're seeing.

Uninstalling

To uninstall PingPlotter - just locate your PingPlotter directory in the start menu and run "Uninstall or Modify Installation." Once the setup wizard launches, click "next" and then choose the "remove" option. On the next screen, you'll be prompted with options to keep or remove your PingPlotter license key; as well as any settings, directories, and log files (note: if you are uninstalling through Add/Remove programs these options are only available through the "Change" option. The "Uninstall" option does not provide these options). After your preferred boxes are checked or unchecked, click "Remove," and then "Finish" to complete the uninstall.

If you can't find the Modify or Remove option in the PingPlotter start menu or in Add/Remove programs, try installing the latest version, then uninstalling. This might also help solve missing installer errors (like "File {7c7a8eba-cb1a-4626-aa2d-8311b84092ab}.msi not found") during uninstall.

4.2 Target Bar



#1 - **All Targets Summary Tab:** The "All Targets" [summary](#) is always available in PingPlotter Pro (only), and list all targets you're tracing to. Super-useful.

#2 - **Trace Target Tab:** Target windows can be opened for any target from a summary screen by either right clicking on a target and choosing "Show Target Window," or by clicking on the IP address for a target.

#3 - **New Trace Tab:** A blank tab ready for a new target entry. If you try to start a second "New Trace", the old "New Trace" is replaced with a new one.

#4 - **Add New Trace Button:** This button will open a "New Trace" window.

#5 - **Target Entry Field:** This field is where a new target can be entered ([either by IP address, or DNS name](#)). Clicking the dropdown button on this field will bring up a history pane showing all of the previous targets that have been traced to.

#6 - **Start/Pause Button:** This button can start a new trace (after a target name/IP address has been entered), or pause an existing trace. Clicking the dropdown option on this button will also provide options to "Reset & Restart" a target (which begins a completely fresh trace session to that target, and clears out all old trace data), as well as an option to create a "New Target" window.

#7 - **Trace Interval Setting:** This setting dictates the amount of time that PingPlotter will wait between sending out sample sets. The dropdown menu provides several different options - but values can be manually typed in here as well.

#8 - **Focus Setting:** This field dictates how much recent data PingPlotter use to calculate the statistics displayed in the trace grid. If set to "auto" - then the focus time will automatically adjust to what is displayed in the time graphs (for example, if a time graph is set to 1 hour, then PingPlotter will automatically display the statistics for the last hour). If set to a value (i.e. - 60 seconds), then PingPlotter will display the most current samples within the timeframe that is set (double-click the [time graph](#) to change focus in non-auto mode). If set to "Auto" - PingPlotter will base its statistics off of whatever is displayed in the time graphs (and the statistics will update if you scroll back and forth on the time graph).

#9 - **Settings dropdown**: If you're using Pro and have more than one [named configuration](#) , you'll have a "Settings" selector here, too. This lets you switch a target (or summary graph) to a different set of parameters.

#10 - **Good/Warning/Bad Color Legend**: This shows the thresholds that have been set for the colors in the graph background. The values here can be adjusted by going to "Edit" -> "Options" -> "Display."

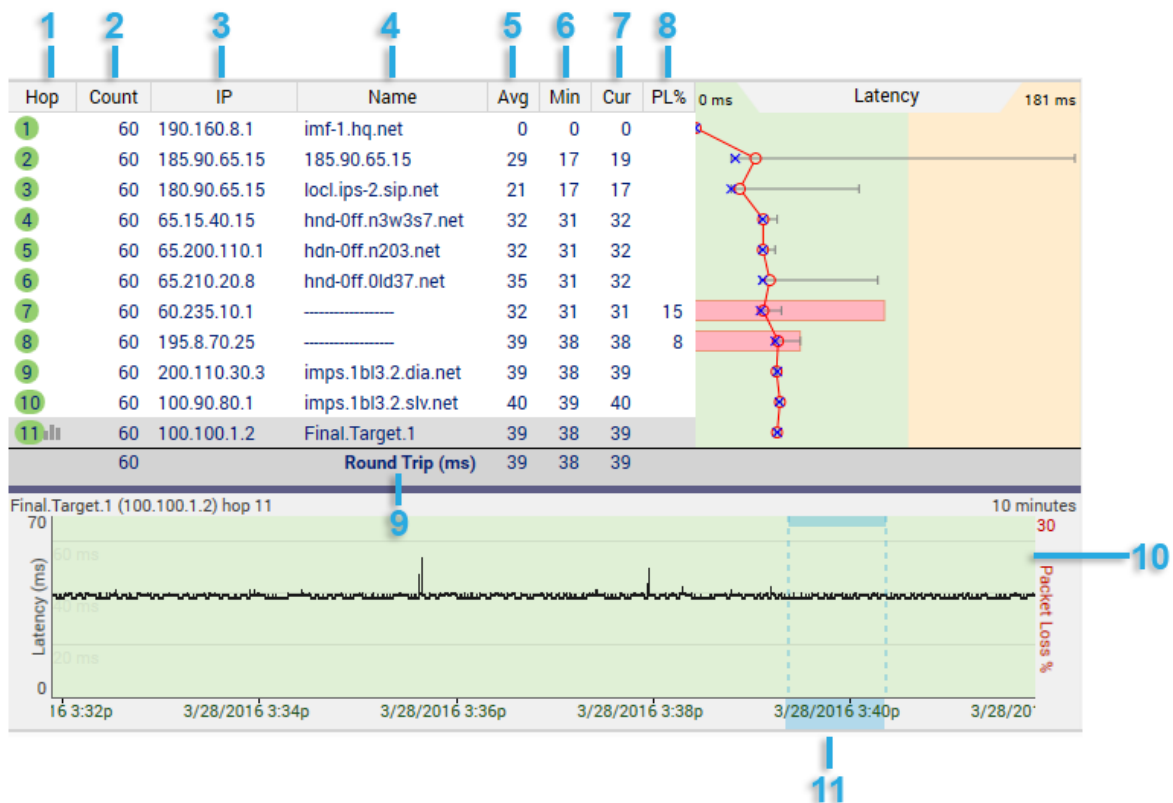
Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details

4.3 Trace Windows

The graphs are where PingPlotter really shines. At a glance, you're able to visually see where a problem lies. There are actually three discreet areas: the **Trace Data Grid**, the **Trace Graph** and the **Timeline Graph**. The first two (Trace Grid and Graph) are referred to primarily as the Trace Graph, as they are used together. Please refer to the below image (annotated with numbers) that we've saved from PingPlotter ("File" -> "Save Image"), and the explanations (referencing the numbers) below the graphs.

The upper graph is called the **Trace Graph**. All columns on the trace graph are re-sizable. The lower horizontal graph is called **Timeline Graph**.

All numbers on the trace graph use the "[Focus](#) " setting to control how many samples are used in the calculations. By default, you're looking at the most recently collected samples - but you can also focus on samples that are not the most recent samples by double-clicking on the time graph. This will focus the upper graph on that period you double-clicked (we cover this topic in a bit more detail below).



- 1 - The **number of hops** that device in the route is from your computer. If a hop has a bell icon next to it, this means that hop is being monitored for an alert (alerts are covered in the advanced settings section of this tutorial). Multiple alerts can be configured for the same IP, and alerts don't work unless some IP in your current route is being monitored. If a hop has a graph icon next to it (like hop #11), that hop is being traced on a time-line graph.
- 2 - The **Count** column displays how many packets have been sent for the current focus period.
- 3 - The **IP address** column shows the IP addresses of the routers that reported back for each hop.
- 4 - This column shows the **DNS name** of the device for that hop. A "-----" in this column indicates that PingPlotter was unable to resolve a name for that device's IP address. This is not a flaw in PingPlotter, it just means that your DNS server doesn't have a name for that IP address (or that address just doesn't have a DNS name, period).
- 5 - The **Avg** column shows the average response time of the last "X" samples (where "X" is the "Focus Time"). Any timeouts/lost packets are not included in this value.
- 6 - The **Min** column shows the minimum response time for the current focus period.
- 7 - The **Cur** column shows the individual sample time of the most recent sample included in the set. If a

number is displayed as **ERR**, that means the packet was lost (i.e.: a packet was sent out, but never made it back).

- 8 - The **PL%** indicates the number of packet(s) that have been lost in the current sample set. If you're only including the last 10 samples, then only the number of lost packets in the last 10 samples are shown here. If you want to find out how many time-outs have happened over the entire session, change the "Focus Time" to "0" or "All."
- 9 - The **Round Trip** line is basically there for ease of reading. It's the same value as the last server in the route. This is the time it takes for a ping to get from your computer to the target device and back.
- 10 - The **Timeline Graph** (TG) is one of the most powerful features in PingPlotter, and great for long-term monitoring projects you may be doing. We discuss time graphs a bit more in the "[Timeline Graphing](#)" section.
- 11 - Double clicking anywhere on the timeline graph will bring up a **Focus Area**, which will focus the trace data graph (top graph) to that particular point in time. This is particularly useful for investigating spikes or time-outs

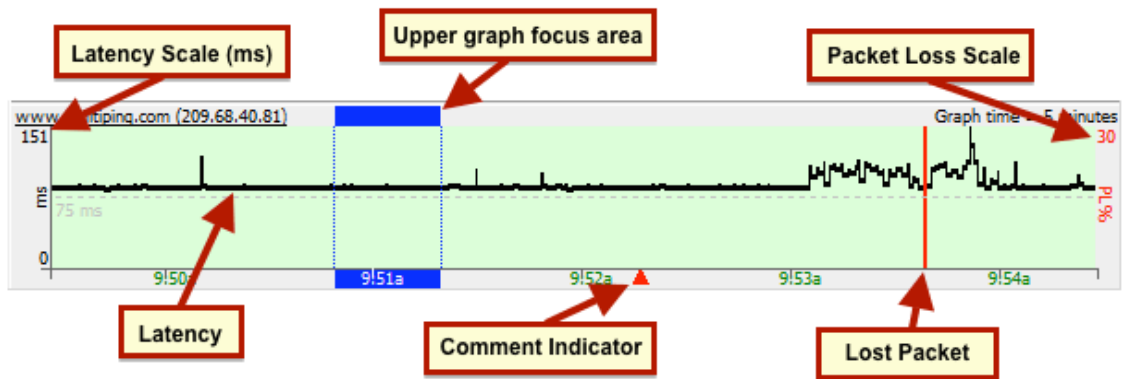
Tools and other options available for the Trace Graph

- You can display the Minimum and Maximum columns by right clicking on the upper trace graph and selecting them (via the "Customize View..." option).
- You can copy the IP address or DNS name for a hop to the clipboard by right-clicking on that hop, selecting the Clipboard option and then clicking on what you want to save.

Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details

4.4 Time Line Graphs

Network problems can often happen when you're not watching for them. The timeline graph feature in PingPlotter gives us a quick way to look over a visual representation of our trace data. This makes spotting problems (or potential problems) much, much easier.



By default, PingPlotter will automatically trace the last hop (the host you're tracing to) on a timeline graph.

In PingPlotter Standard and Pro you can also display a timeline graph for any of the other hops in a route by either double clicking on that hop, or right clicking and selecting “Show this timeline graph.” You can also turn off any graph by these same mechanisms.

The amount of data displayed on the graph can be changed, too - just right click anywhere on the graph and select the amount of time you'd like to display (this will affect all timeline graphs and is saved when you shut down PingPlotter).

Navigation

If there is more data collected than we can show on a timeline graph (for example, if you've got 48 hours worth of data, but have your graph scale set to "10 minutes"), you can click (and hold down) your mouse button on the graph, and drag it back and forth. This allows you to move back in history and examine the samples during those times.

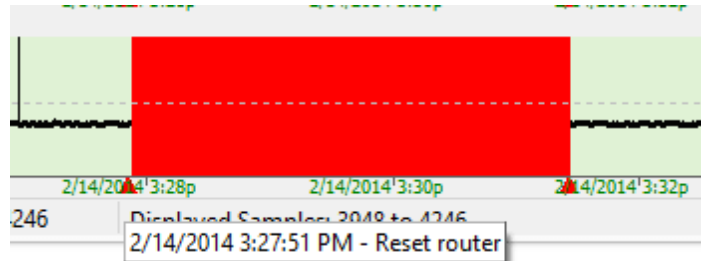
The scroll wheel on your mouse also works some magic with the timeline graphs. If you click on a graph and scroll down, you'll move back on the time graph, scrolling up will move the graph forward. If you click the scroll wheel, scrolling up or down will toggle through the time period options on the graph (so you can basically zoom in, or zoom out using this method).

Comments

If you're using PingPlotter for long-term monitoring, you may sometimes run across a situation where your network was effected by something you know about (power outage, big file download, that time you threw your router out the window, etc). You may also run into situations that you don't control, but know the cause of (or can speculate on). Being able to take notes about these situations and tie it to your data

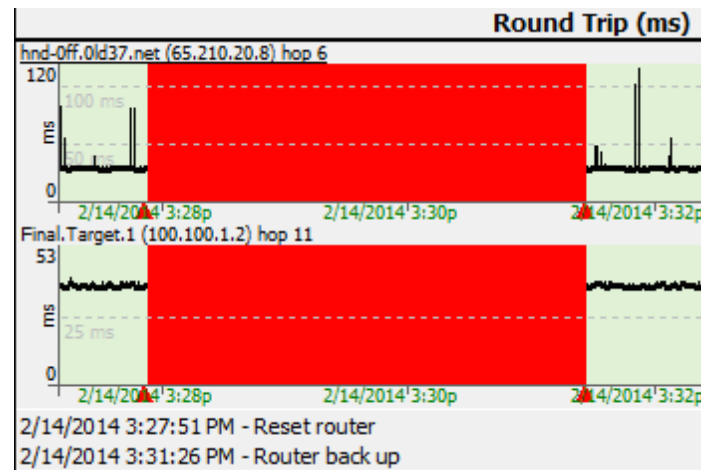
can prove to be very, very helpful.

To create a note, all you need to do is right click on a timeline graph (at a point you want to create a note), and select the “Create Comment” option. A prompt will appear asking you to create your comment, and then PingPlotter will draw a red triangle on the lower edge of the timeline graph. If you float your mouse over the triangle you can see the note:



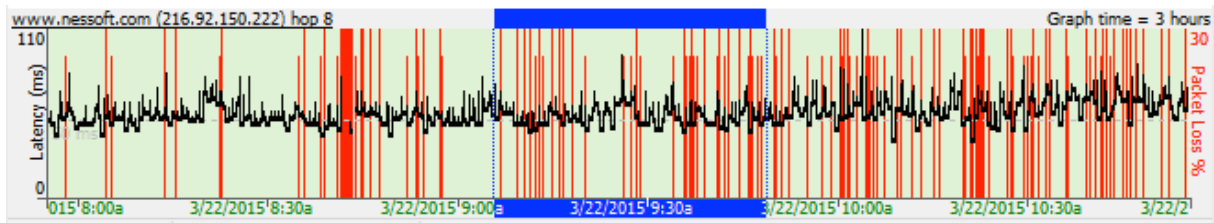
Right clicking on the comment triangle will also allow you to edit or delete it.

When you save data as an image from PingPlotter, your comments will get attached to the image (along with the times they happened). If you're sending a image to a network provider, this can be especially valuable - as it helps explain the events on the data that you're sending to them:



The Focus Area

Any time you double click on a timeline graph, a blue “Focus Area” will appear, which focuses the upper graph to that point in time. This focus area is based off of the “Focus Time” value (and this won't work if you have this value set to “ALL”).



If you've got your graphs set to show 48 hours worth of data, and if you find a time period that looks like it might be interesting, you can double click on the timeline graph at that point and the trace graph will move to that time period as well. You can then change your timeline graph scale, and the lower graphs will stay focused on the period you selected. This makes it easy to spot, and zoom in on problems.

When you're finished going through your graph history, you can reset everything to display your current results by right clicking on a timeline graph and selecting "Reset focus to current." This returns both of the graphs to, you guessed it, the current time.

Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details

4.5 Summary Graphs

The summary graph screen in PingPlotter is a listing of any final destinations or intermediate hops from any of your destinations. This summary gives you a way to compare multiple targets, and also gives you an easy one-glance summary of all the important parts of your network.

The "All Targets" summary

The *All Targets* summary is a special summary. It's all the final destinations you're tracing / monitoring.

This summary is automatically maintained by PingPlotter - when you add a target it shows up, when you close, it's removed. You can rename the tab (and any other summary) by right-clicking on its tab, but it's always maintained. If you hide, it'll be there (in the [Workspace](#) menu) when you're ready for it again.

Adding targets to the summary screen

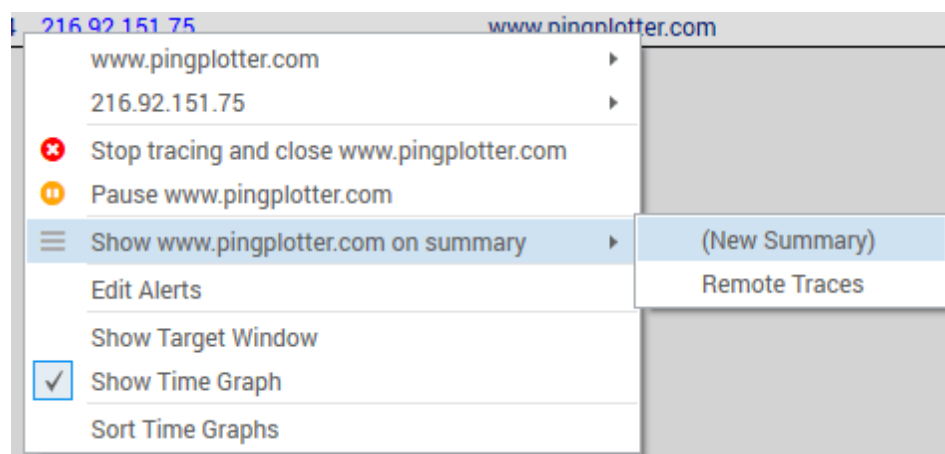
If you want to manually add a target (or any intermediate hop) to a custom summary graph (which we cover below), just right-click on that hop in the trace graph and select "Show (target) on summary" -> (then pick a summary).

From the summary screen, you can switch to the trace graph by right-clicking on a host and selecting the menu option "Show trace graph," or by simply clicking on target's IP address. You can sort the list of

targets by clicking on a graph heading. Clicking again reverses the sort order.

Creating custom summary screens

You can create multiple custom summary screens, making it easy to group certain targets together. To create a new summary screen simply right click on a target (or you can select multiple targets by holding down the "CTRL" key and clicking on them, and then right click) and under "Show on Summary Screen" select "(New Summary)" and a new tab will open up with your target(s).



You can remove targets from a summary screen by right clicking and selecting "remove from summary."

Summary screens that have been hidden can always be reopened by going to the "Workspace" -> "Summaries" menu option - which will display a list of any active summary screens that are set up. If you want to auto-maintain a summary screen based on performance, you can use an alert to add or remove from a summary.

Time periods on the summary screen (or, "Focus")

A summary screen is a collection of targets, possibly with different trace intervals, latencies ranges or problem / performance characteristics. The "Focus" always applies the same time period to all targets, even if the trace interval is different. Some targets may be paused while others are running - they're all compared at the same time, so your statistics may be blank for some targets, and your "Count" column may have different numbers for each target.

As in the Trace window, the "Auto" focus mode is pretty awesome - you can scroll through time and get a quick summary of what you're looking at.

Using the Target Bar on a summary screen.

A great way to enter new targets is to do it from a summary screen. When you do this, the settings and

interval from the summary target bar is used - and that target is automatically added to that summary screen. When you do this, a target tab is not opened for each target you're adding, but the data is being collected. Just click on the blue IP address link to open up the full trace graph.

Another cool thing about the target bar is you can easily change interval or settings - just select a target (or 10 via multiselect) and then change the setting.

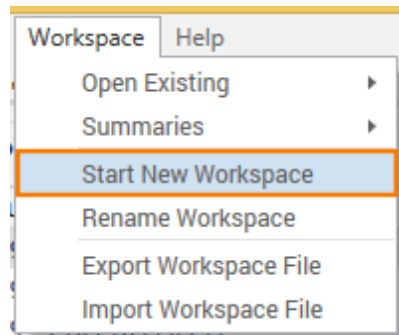
On a related note, multi-selected targets have a variety of options available on the right-click menu, too. You can close, start, stop, etc all the listed targets; the right-click menu will show you the affect you're going to have.

The feature in this topic is exclusive to PingPlotter Pro. See our [product comparison](#) page for more details

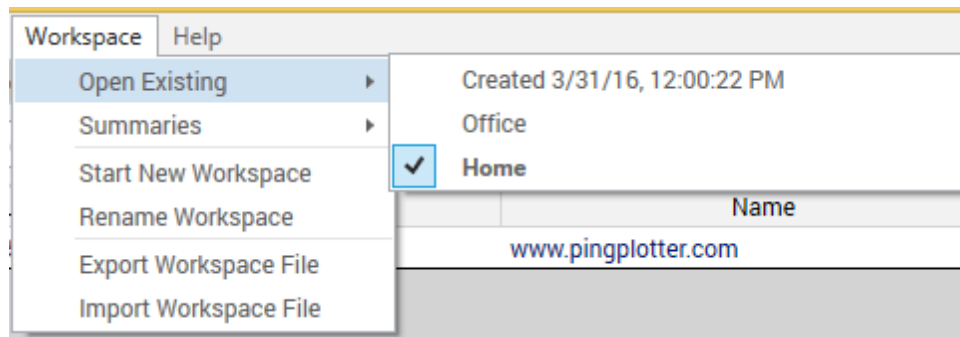
4.6 Workspaces

Workspaces capture information about your setup; including the targets that are being monitored, the configuration each target is using, trace intervals, focus times, and screen and summary layouts.

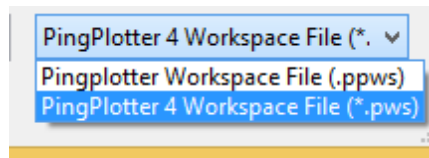
PingPlotter v5 always has a workspace running by default. If a separate workspace is needed (for example, if we'd like to set up a "home" and an "office" workspace), one can be created by going to "Workspace" -> "Start New Workspace."



Workspaces are automatically given a generic name ("Created" followed by a date/time stamp of when it was created), but can be renamed by choosing the "Rename Workspace" option in the Workspace menu. All workspace information is saved to the PingPlotter database on a machine, so any workspace can be opened via the "Open Existing" option:



Older workspaces from v4 versions of PingPlotter can be imported via the "Import Workspace" option. When browsing for the workspace file, the option for "PingPlotter 4 Workspace File" needs to be selected from the dropdown menu in the lower right hand corner:



Workspaces can also be exported if needed. It's important to note that in PingPlotter v5 these files are saved as .ppws files - so any workspaces saved from v5 will not be compatible with older versions of PingPlotter.

Automatically loading a workspace on startup

When you launch PingPlotter, right after installing, it will create a new workspace every time you launch it. When you close and restart it will launch a *different* new workspace. This lets you launch a target, trace for a bit, and then close. Next time it's fresh and ready to go with your next task.

The workspace you used last time is automatically saved, though (see [auto-save settings](#) for more details on when it might not be), and if you want to continue your last project, just open that workspace from the load screen (or workspace menu). Set it to auto-open if you always want to start tracing when you start PingPlotter (or if you want the service instance to do that).

Once a workspace is set to start automatically, you can set any target to stick around or to get thrown away when you close it. Right-click on a tab to change modes. This lets you collect data to a few targets all the time, but still create an occasional "adhoc" session to do spot troubleshooting without having that stick around forever. The workspace will remember your preferences here, and reload (or not) the next time the workspace is reopened. More details on this is discussed in the context of [auto-save](#).

The [Windows service](#) and workspaces

When running as a Windows service, there are some complications with workspaces if you're in a multi-

user environment. On startup, the Windows service automatically starts, and any user on that machine can close targets, switch workspaces, or add new targets.

The list of running targets is stored separately from the "layout" of a workspace. The list of targets is tightly tied to the data [we're collecting](#) for those targets, and will always win over your layout workspace. It's possible that starting PingPlotter will connect to a different list of targets than last time - in this case, PingPlotter will do its best to give you something meaningful (like a nice [All Targets](#) summary) - and your collected data will be there, read to review and analyze.

The feature in this topic is exclusive to PingPlotter Pro. See our [product comparison](#) page for more details

4.7 Web Interface

PingPlotter Pro has a built in web interface, which allows you to view data and access quite a few options of PingPlotter Pro from a web browser - without having to be local on the machine that is collecting data.

Enabling the built-in web interface

We recommend using PingPlotter as a Windows Service if you're going to use the web interface. This will make sure it's always running, and makes security permissions slightly easier to configure.

To enable the web interface, go to "Edit" -> "Options" -> "Web Server" and turn on the web server (yes - it's seriously that easy).

We recommend leaving the port at 7464 (although you're free to change it) - and if you're doing some local testing, it's best to turn off the "Use Security" checkmark (once your testing is finished, you can go back and re-enable this option). Apply your changes, and close down the options dialogue.

We want to make sure that the web interface is running correctly - so in PingPlotter Pro, make sure you're tracing to at least one target, and then move on to the next section.

Accessing the web interface

To access the web interface from your local machine, fire up your favorite web browser (if you're wondering about browser requirements, we cover that further down this page) and enter the following address (or click it, if you're reading this on your computer):

<http://localhost:7464>

This should open your browser to the PingPlotter Pro web interface - which will list all of your targets. You can click on a target to open a trace graph for it, stop tracing (and close down a target), add a new target, change some basic settings, and do some basic examination of data.

Doing Basic analysis

You can do quite a few things from the web interface - but *some* actions do require the local PingPlotter Pro client. Wondering what's possible to do with the web interface? Well, we've compiled a list:

- Add/remove targets
- Create a custom summary screen (and add/remove targets from it)
- Add/remove columns, customize widths.
- Examine a target's data by:
 - Changing the focus period
 - Changing the period of time on the timeline graph
 - Turn on/off time graphs
 - Download sample data for more analysis within a local instance of PingPlotter
 - Do whois lookups through www.whois.sc

Sharing targets between the Windows GUI and the web interface

Good news here - everything "just works". If you're running as a service, booting your machine will start the web interface and load a workspace (based on your preferences). You can use the main PingPlotter GUI to create alerts, add/remove targets, create summary screens. Your changes will be reflected on the web interface (which is a big improvement from V4 of PingPlotter). This means someone accessing the web interface can close targets created on the GUI and create targets that show up on the GUI.

Named configurations need to be created with the GUI, but then will show up on the web automatically.

Browser Requirements

PingPlotter 5's web interface requires that Javascript be enabled. We recommend using at least IE11, Firefox 10+, or any other browser that is less than a couple of years old.

For PingPlotter v4 web interface users

Using IIS as a web server

PingPlotter 4's web interface used Classic ASP with VBScript, which is now pretty old and doesn't make sense to duplicate. The good thing about this tech is that it worked OK with IIS as a backend web server.

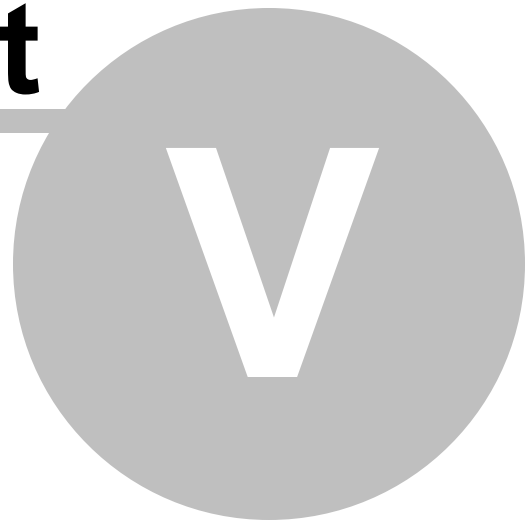
PingPlotter 5 uses a completely different architecture - one that does not work with VBScript or Classic ASP. Currently, it also doesn't work with IIS. If you need IIS (for security, stability, scalability reasons), you might want to keep using PingPlotter V4. If you're using IIS so you can use https, that is coming to PingPlotter 5 (and should be possible today, although not documented or tested). Support for other web servers (IIS, Apache on Mac, and others on other platforms) is on the roadmap but is not immediately possible. Please contact us and let us know how you're using PingPlotter web and we'll help come up with best solution.

Advanced configurations

Over the years, we know our users have customized the web interface to do things we didn't really design PingPlotter to do - and we've helped with some of this. PingPlotter 5 changes what's possible here. If you're using V4 with modifications to the web interface to allow read-only views, REST calls to add targets, surfacing custom data and similar, please contact us for help. Many things are possible, but it's different than V4.

The feature in this topic is exclusive to PingPlotter Pro. See our [product comparison](#) page for more details

Part



Common Tasks

5 Common Tasks

5.1 Alerts

5.1.1 Creating / Configuring Alerts

What is an alert?

Alerts basically monitor the conditions of a specific IP address, and then do something when those conditions exceed a specific range. The things you can do with an alert are:

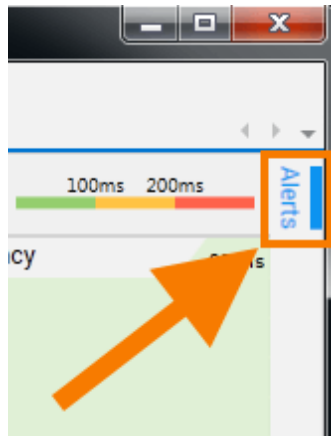
- [Play a sound or .WAV file](#)
- [Send an email](#)
- [Log to a text file](#)
- [Change the tray icon / show a message](#)
- [Execute a web REST call](#)
- [Add/Remove a target from a summary screen](#)

For example, let's say you need to know when a destination you're monitoring stops responding. You can attach an alert within PingPlotter to that IP address so that you receive an email alert if the last 10 of 10 sample requests are lost.

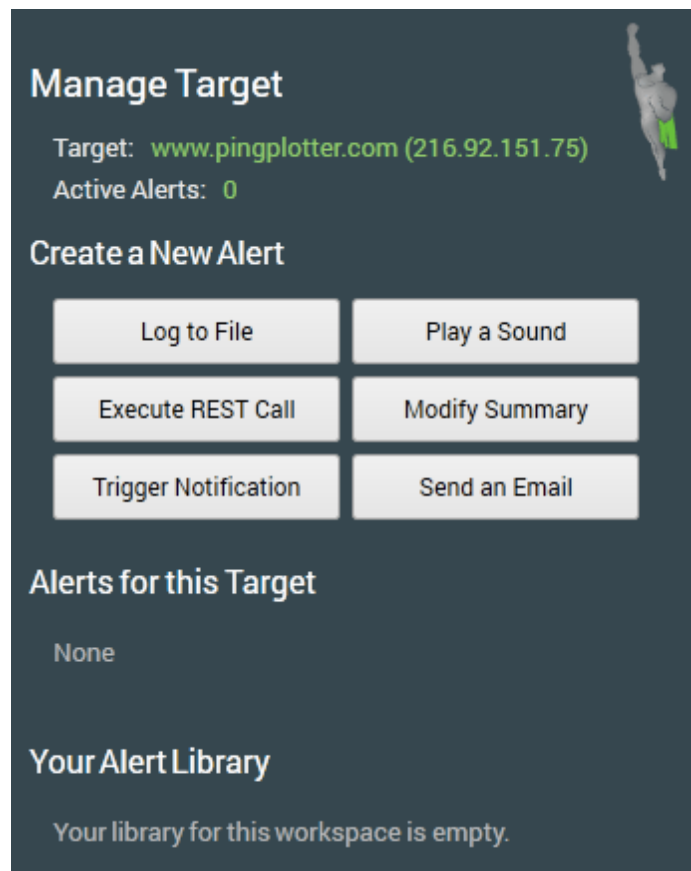
Another possible alert condition to check for is if the average for the last 10 samples is > 500 (or any other number). You can send an email alert, maybe play a .WAV file (if you're usually within hearing distance) or both. Also, if you're trying to show your ISP there's a problem, you might log the data to a file so you have records of every time it happened over a time period.

Setting up an alert.

PingPlotter v5 introduces a new guided alert setup process. To start setting up a new alert, either select a target from a summary screen, or open a trace window for a target, and click on the "Alerts" tab - which can be found on the right hand side of the application.



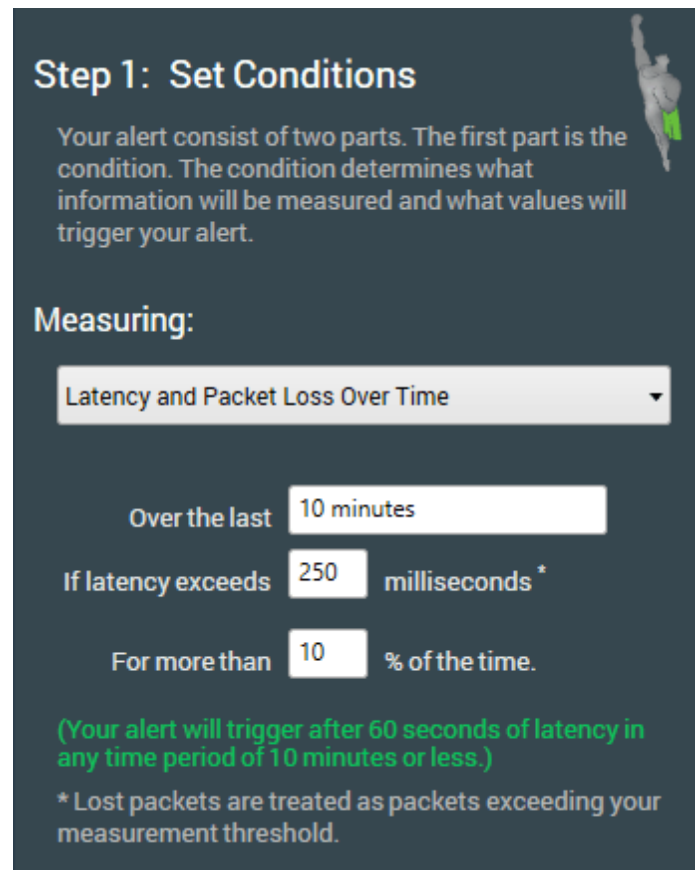
Once the alert sidebar is opened, we land on a screen that will tell us any alert info that may be present for the target we've chosen. From here, we're also be able to create a new alert that will be stored in our alert library - so that we can use it on any of our targets.



Setting up a New Alert

For this example, we'll be setting up an alert that will play a buzzer sound anytime our target experiences latency higher than 250 milliseconds for 10% (or more) of a 10 minute timeframe. First, we'll

select the "Play a Sound" option from the "Create a New Alert section, which will prompt us to enter our alert conditions.



Step 1: Set Conditions

Your alert consist of two parts. The first part is the condition. The condition determines what information will be measured and what values will trigger your alert.

Measuring:

Latency and Packet Loss Over Time

Over the last 10 minutes

If latency exceeds 250 milliseconds *

For more than 10 % of the time.

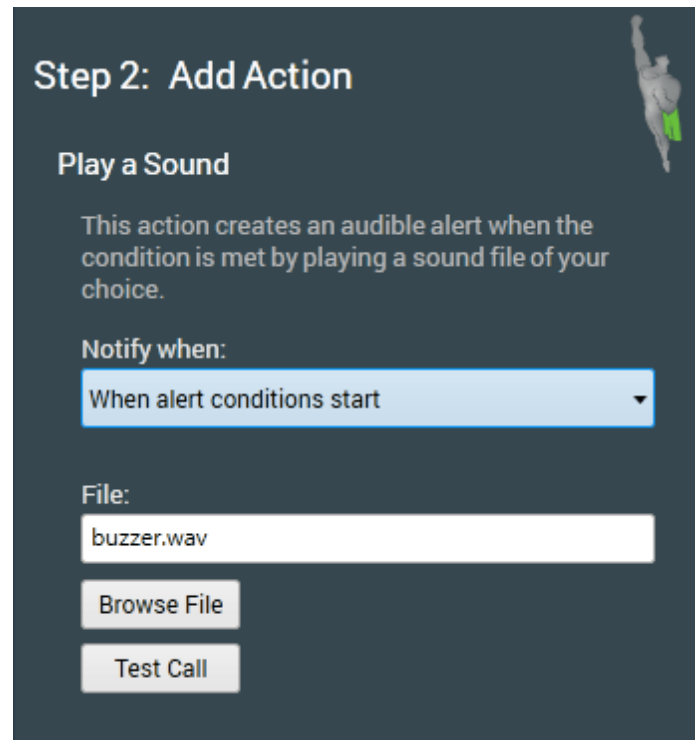
(Your alert will trigger after 60 seconds of latency in any time period of 10 minutes or less.)

* Lost packets are treated as packets exceeding your measurement threshold.

There are two different options for setting up alert conditions:

- Latency and Packet Loss Over Time - (the option we're using in this example)
- Latency and Packet Loss Over a Sample Count (which allows us to look at a certain number of samples, and trigger an alert when a set amount of samples exceed our desired latency)

Once we've set our conditions, click on "Next," and in step 2 we can configure what sound file we'd like to play, and when we'd like it to play.



Step 2: Add Action

Play a Sound

This action creates an audible alert when the condition is met by playing a sound file of your choice.

Notify when:

When alert conditions start

File:

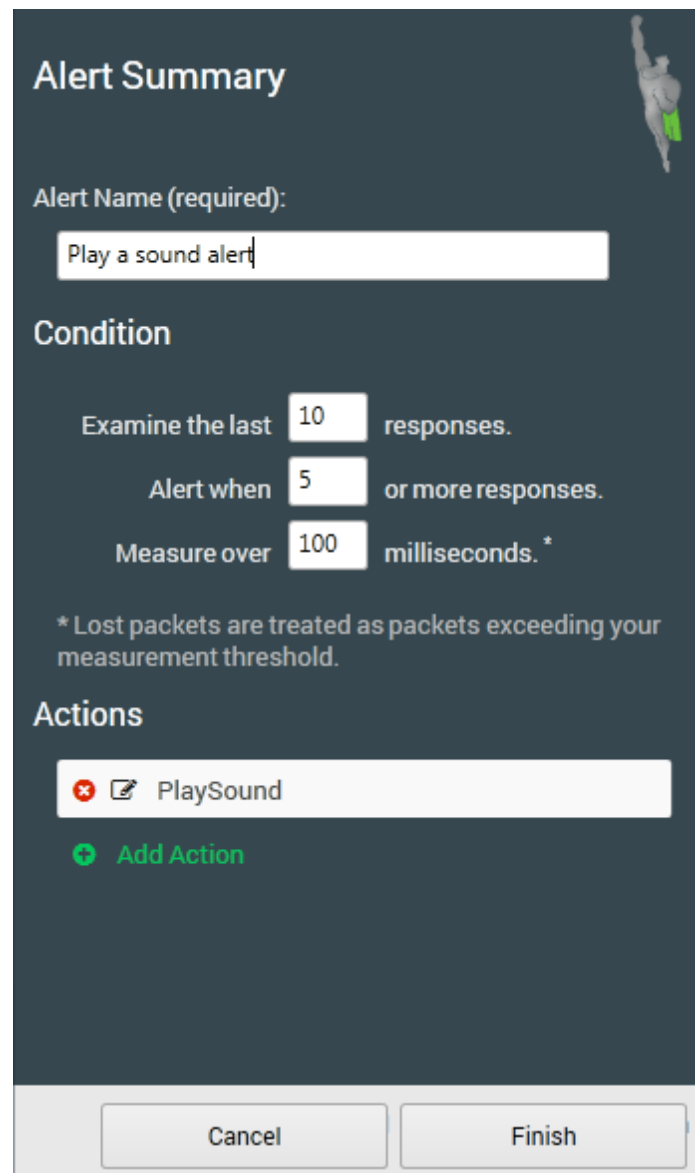
buzzer.wav

Browse File

Test Call

There are four different notification preferences:

If we'd like our sound to play when alert conditions start, and then have a different sound play when the conditions end, this is totally possible (and encouraged!). We'll go ahead and select "When alert conditions start" for the first part of this alert, and then choose the sound file we'd like to play and click "Next."

The image shows a software interface titled "Alert Summary". In the top right corner, there is a small icon of a person in a white suit with a green shield. Below the title, the text "Alert Name (required):" is followed by a text input field containing "Play a sound alert". Underneath this is a section titled "Condition". It contains three rows of configuration: "Examine the last" followed by a numeric input field with "10" and the text "responses."; "Alert when" followed by a numeric input field with "5" and the text "or more responses."; and "Measure over" followed by a numeric input field with "100" and the text "milliseconds.*". Below these is a footnote: "* Lost packets are treated as packets exceeding your measurement threshold." The next section is titled "Actions" and contains a list item with a red 'x' icon, a pencil icon, and the text "PlaySound". Below the list is a green circular button with a plus sign and the text "Add Action". At the bottom of the screen are two buttons: "Cancel" and "Finish".

Alert Summary

Alert Name (required):

Play a sound alert

Condition

Examine the last 10 responses.

Alert when 5 or more responses.

Measure over 100 milliseconds.*

* Lost packets are treated as packets exceeding your measurement threshold.

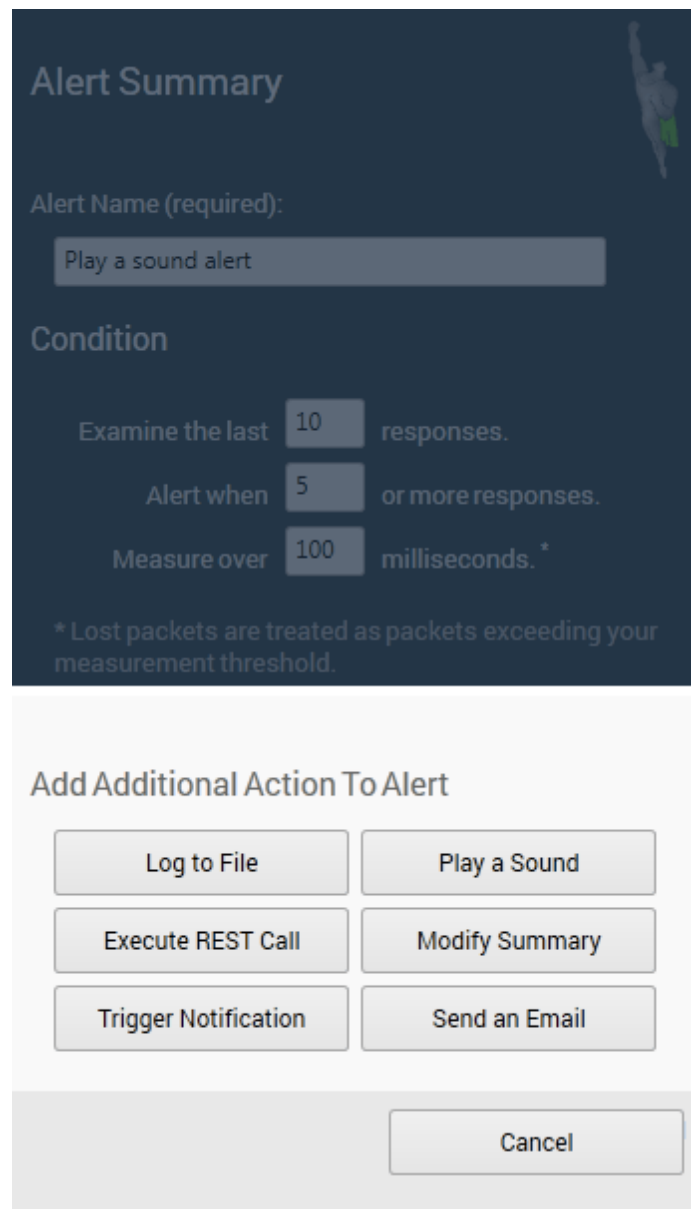
Actions

✖ ✎ PlaySound

➕ Add Action

Cancel Finish

From the alert summary screen, we'll get an overview of what our alert is configured to do. To set up another sound that will play when alert conditions end, click on the "Add Action" button.



Alert Summary

Alert Name (required):

Play a sound alert

Condition

Examine the last 10 responses.

Alert when 5 or more responses.

Measure over 100 milliseconds. *

* Lost packets are treated as packets exceeding your measurement threshold.

Add Additional Action To Alert

Log to File Play a Sound

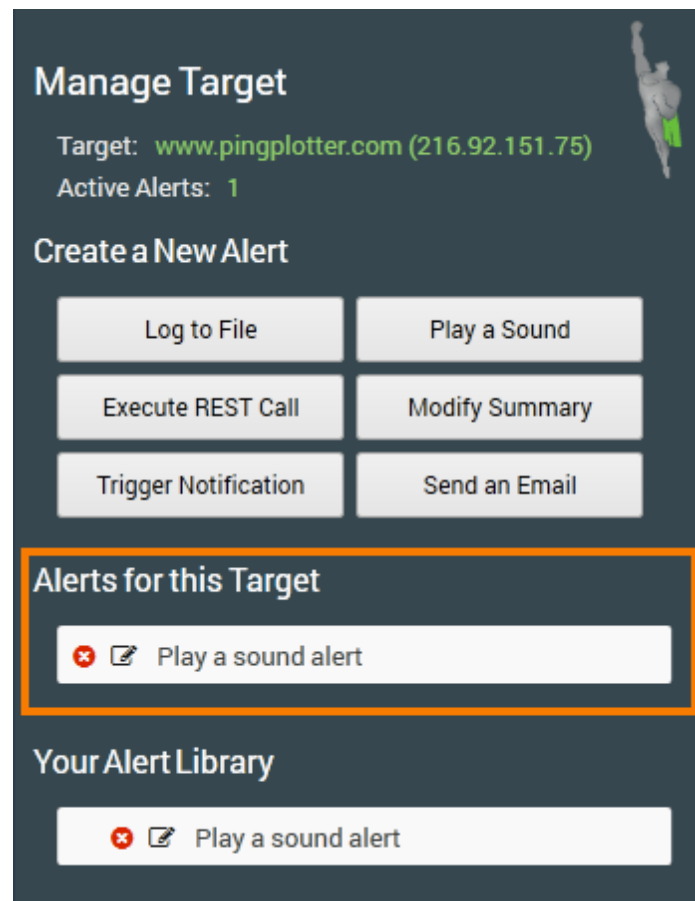
Execute REST Call Modify Summary

Trigger Notification Send an Email

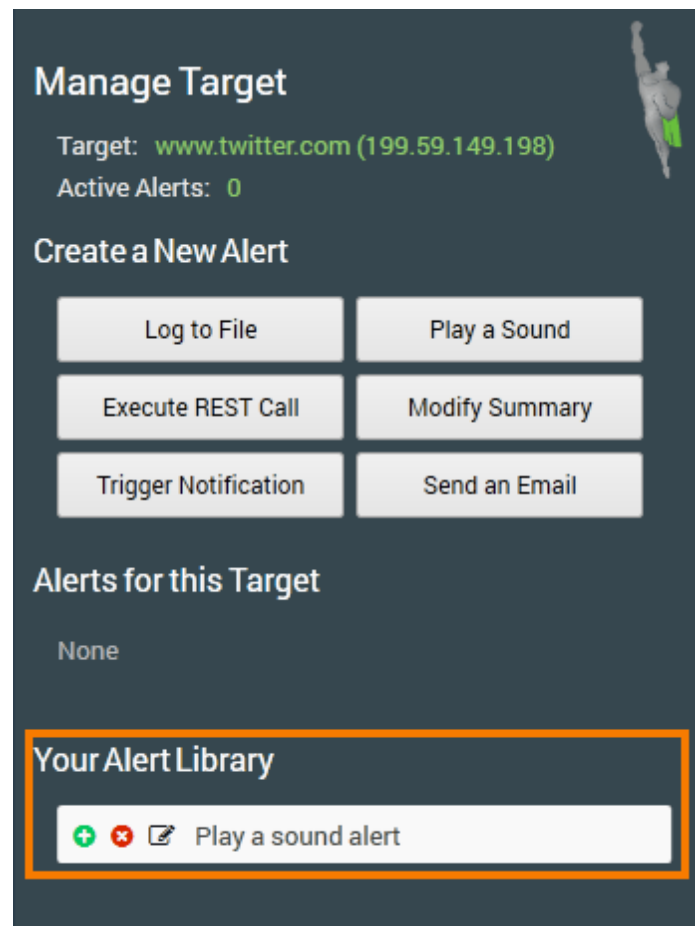
Cancel

From this screen, we can select another action type and set the notification preference we'd like ("When alert conditions end"). Once everything on the alert summary screen looks to be in order - we can give our alert a name and click on the "Finish" button.

Once we've finished setting up our alert, we'll land back on the original alert screen, which will now show our new alert under the "Alerts for this Target" section. If we want to change anything about the alert (pick a different sound file, adjust the alert conditions, etc), we can click on the edit button. We can also remove the alert from a target by clicking on the delete button. Clicking on the delete button in the "Your Alert Library" section will completely remove the alert from the program (as well as from any targets it happens to be associated with).



If we've got another target that we'd like to apply our alert to, simply open a trace window for that target. The alert will be in the alert library, and clicking the green add button will move it into the "Alerts for this target" section.



Alerting on packet loss.

A common condition to want to alert on is packet loss. The fields you need to manipulate are in the Alert setup screen is the "Alert conditions" portion.

Example: Let's say you want to notify when packet loss equals or exceeds 40%.

To do this, set "Samples to Examine" to 10, and Alert when "4" or more samples are over 9999ms. A lost packet always exceeds any number you enter in the threshold area, so if you want to consider only explicitly lost packets, set this to 9999. If you want to consider any really high latency packets as well, set this to something lower (maybe 1000 or 1500).

This only examines the last 10 packets, but let's say you want to examine a higher period - and notify on a lower packet loss percentage.

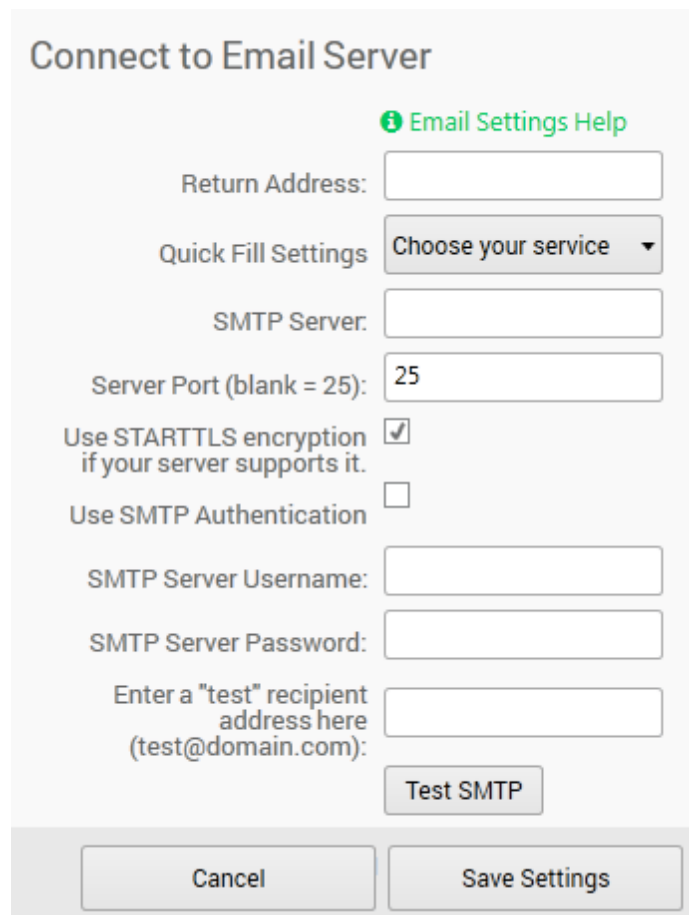
Set Samples to examine to 10000 (or some other high number). Alert when "500" or more samples are over 2500ms. This will alert when you hit 5% packet loss over a period of a few hours (depending on what trace interval you use).

A note about "average" response times.

Average response times are a problem. The real problem with **mean** averages is when a server stops responding - what is the average of the last 10 samples if the last 10 were timeouts? Because of the problem with this we always do "when X or more samples is > Y" (this is a **median** average). You can still get your alert to work like an average - by saying "when 5 or 10 samples exceeds 300 ms" (this would be like a mean average over 300ms, but would also fire when there were lost packets).

5.1.2 Email Setup (for alerts)

The process of setting up an email alert is mostly the same as setting up any other alert, with a few exceptions. The first big difference is that we'll be asked to configure email settings before setting up any other part of the alert.



The screenshot shows a configuration window titled "Connect to Email Server". At the top right is a green link "Email Settings Help". The form contains the following fields and controls:

- Return Address:** A text input field.
- Quick Fill Settings:** A dropdown menu currently showing "Choose your service".
- SMTP Server:** A text input field.
- Server Port (blank = 25):** A text input field containing the value "25".
- Use STARTTLS encryption if your server supports it:** A checkbox that is checked.
- Use SMTP Authentication:** An unchecked checkbox.
- SMTP Server Username:** A text input field.
- SMTP Server Password:** A text input field.
- Enter a "test" recipient address here (test@domain.com):** A text input field.
- Test SMTP:** A button located below the test recipient field.
- Cancel** and **Save Settings**: Two buttons at the bottom of the window.


If we're using a common email carrier (Gmail, Outlook, Yahoo, etc), we can use the "Quick Fill Settings" dropdown to automatically enter the correct SMTP server name, and port. Once we've filled everything here out, we can send a test message to make sure we've gotten everything correct - we'll just enter our email address in the last field on this screen (where it says "enter a "test" recipient address here") and

click on the "Test SMTP" button. If all goes according to plan, we should receive an email in our inbox letting us know that we're good to go!

On the screen for step 2, there is an option to edit our email setup if we need to make some adjustments (just click on the "Email Setup" icon).

Next we'll need to set up our notification preferences, and choose the address (or addresses) we'd like to send the alert email to. If we need to send the email to multiple people, we can enter multiple addresses in the "email recipient" field, each one separated by a comma.

Step 2: Add Action

Send an Email  Email Setup

This action emails a snapshot of the response values for each of the hops to your target to the recipient or distribution list you specify.

Notify when:

When alert conditions start

Email Recipient

Subject Template

{{Host.PreferredName}} Down!

Message Template

```
Alert "{{AlertName}}" fired for
{{Host.FullDisplayName}}
{{attach 'pp2' of_last='60 seconds' samples=100}}

```

Edit Template Test Email

Cancel Next

We can also edit the body of the email that's sent by clicking on the "edit template" button. This will launch a template editor that will provide several different options for what to include in the body of the

email.

☒ HTML ☐ Text

Subject:

```
Alert "{{AlertName}}" fired for {{Host.FullDisplayName}}
{{attach 'pp2' of_last='60 seconds' samples=100}}
 "View Alert Events Log") provides a summary of any recent alert activity. This prompt will display the date/time of any alert occurrence, the target the alert fired for, as well as a description of the alert. This can be a great help in troubleshooting alerts, as it allows you to see if/when an alert attempted to fire.

### **Make sure the alert is tied to the correct target**

By far, the most common reason that an alert isn't working is because it isn't tied to the correct IP address. If you suspect this may be the case, select the target you're troubleshooting the alert for, and open the alert panel. Here, you should see any active alerts for the target (and if there are no active alerts, you can add them from the alert library from here).

### **Set up an alert that will fire instantly, with an event that is very evident.**

If you have an alert set up – and tied to a host (see above), but it seems like the alert isn't working, then

changing your alert parameters (or create a new "test" alert). Set up "Traces to Examine:" to 10. Alert when "1" or more traces are over 1ms. Unless your network is responding in 1ms or less, this alert will fire on the first collected sample with the alert enabled.

For an event type, use "Play a sound", or "Tray icon change/notification" as both of these events happen immediately with no wait. In addition, for the "Play a sound", use "each time alert conditions are met (repeating)", as this will continuously make sound, rather than just when conditions start / stop. Using this sequence, you should be able to tie an alert to just about any host and have the alert conditions fire immediately. Now, add on another event type (IE: email). You can leave multiple events tied to a single alert – that way you can continue to hear the sound while you're troubleshooting another event type.

## 5.1.5 Alert Events

### 5.1.5.1 Play a Sound

One of the most simple event types is to play a sound (i.e.: .wav or mp3 file) of some kind.

This event can happen based on the [standard notification rules](#), and can play anything that Windows multimedia sound function wants to play.

Clicking on the "Browse file" option when setting up this alert will lead to a default directory, which contains a few sound files that are included with the installation of PingPlotter. Clicking on the "Test Call" button will play the sound file that has been selected.

### 5.1.5.2 Send Email

A very popular event type is the "Send Email" event.

Before you can create an event to send an email, you must [configure your SMTP server and return email address](#). Note that your SMTP server must be accessible on the network to be able to send emails, so it's possible a network failure may not be able to email you. PingPlotter will continue to try to send emails once a minute until it is able to get an email out.

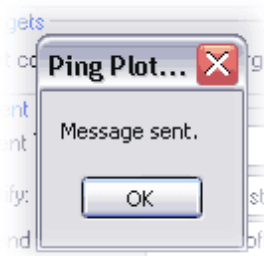
Emails are a bit more complicated to set up than most Event types – as it is dependant on your SMTP server, and you don't want to be overwhelmed with emails when conditions are bad, but you \*do\* want to know what's going on.

First off, you can fire emails based on the standard PingPlotter notification types. [See the associated documentation for more details](#).

## Send e-mail to:

This can be an individual email address, or a list of addresses separated by either a , or a ; (both work equally well). Please do not set this up to be someone at your ISP unless they have agreed that they want to see this information. A huge portion of getting problems solved is playing the game right, and overwhelming people with automated emails is almost certainly going to work against you.

## Testing and error messages



Once you have your e-mail set up, use the "test" button to see what the message will look like (and also to make sure all the settings are working). Any errors should be displayed here.

Many of the errors that occur during testing can be attributed to incorrect email setup – [so go there and validate your settings](#).

Here are some specific knowledge base articles on possible error numbers:

[Socket error 10053](#)

[Socket error 10060](#)

[Socket error 11004](#)

Any Socket error is being generated by the SMTP server itself, not PingPlotter, so if you're getting an error number not listed here, or in [our knowledge base](#), try doing a search on your favorite search engine to see if you can find more information about the error you're getting.

### 5.1.5.3 Log to file

The "Log to file" alert writes data to a text file whenever alert conditions are met.

## Filename:

The filename is required to have a full path specified (for example: "C:\ppdata\Alert.log") in order to save the log file properly.

### 5.1.5.4 Tray Icon Change

This is a great event to add to most of your alerts. It's helpful to be able to see if there's an alert condition under way, and a quick glance at the tray can let you know by using the event.

The Tray Icon Change notification will do one or both of the following:



## Change default icon to red during alert conditions

If you already have PingPlotter showing in the tray, this will change the existing green icon and add red to indicate that an alert has fired. If you don't have PingPlotter in the tray already, then a red icon will be added to the tray. When the alert condition(s) are over, the icon will change back to blue.

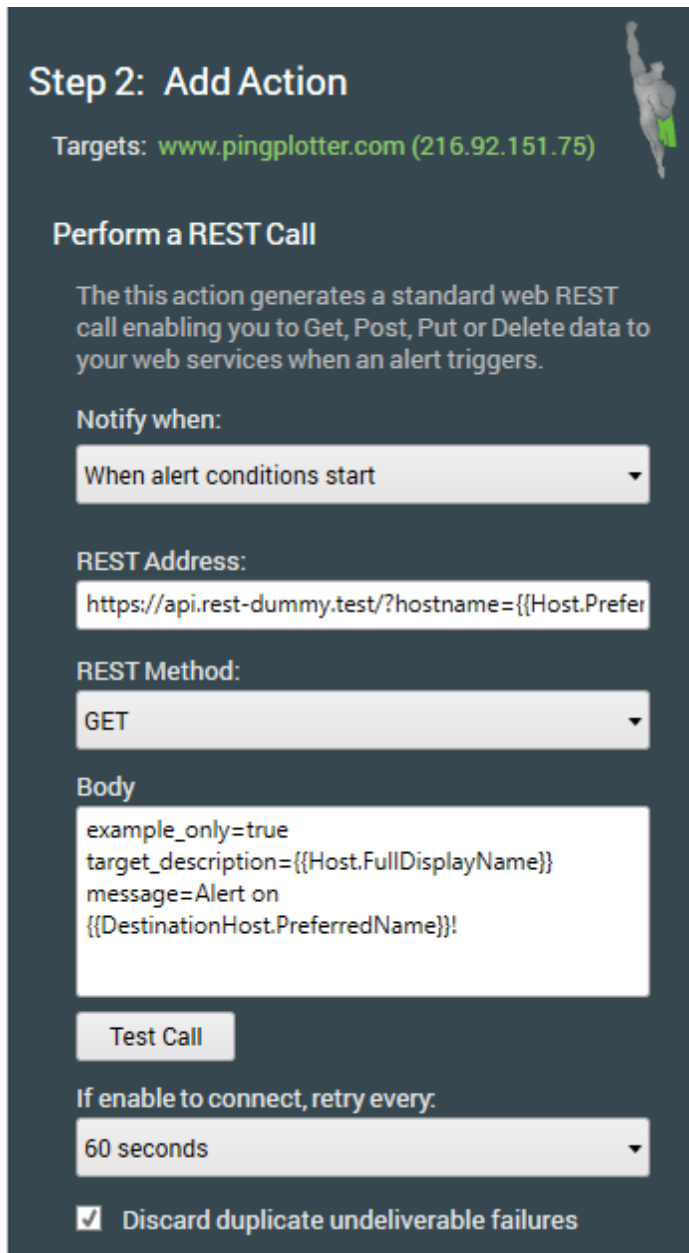
## Popup message in tray

This shows a "balloon" message coming out of the tray. Not all versions of Windows support this message (ie: some versions of Windows 95), in which case no balloon will show. Only one balloon can be shown at a time, so the newest balloon always wins (a new balloon message will replace an older one).

### 5.1.5.5 Web REST Call

The Web REST call event gives you the capability to fire an alert to any service or program that has the ability to receive REST calls.

When you create a Web REST alert event for the first time, you'll notice that there's a "place-holder" example already in place:



**Step 2: Add Action**

Targets: **www.pingplotter.com (216.92.151.75)**

**Perform a REST Call**

The this action generates a standard web REST call enabling you to Get, Post, Put or Delete data to your web services when an alert triggers.

Notify when:

When alert conditions start

REST Address:

https://api.rest-dummy.test/?hostname={{Host.Prefer

REST Method:

GET

Body

```
example_only=true
target_description={{Host.FullDisplayName}}
message=Alert on
{{DestinationHost.PreferredName}}!
```

Test Call

If unable to connect, retry every:

60 seconds

☒ Discard duplicate undeliverable failures

You'll need to have the specific address for whatever service or program you're wanting to fire the alert through here. \*Most\* services will have some sort of documentation on where to find this address, and what needs to be included in the body (i.e.: a security token) in order for things to work.

#### 5.1.5.6 Add/Remove from Summary

This event type is gives you the capability to place (or remove) targets into custom summary screens based off of their performance (which you can specify when entering your alert conditions). To set up this alert event, you'll want to be sure to have a [custom summary screen](#) already set up.

One of our favorite setups with this alert type is to have two events: one that will add a target to a custom summary, and another that will remove it. Using this alert, anytime a target starts to experience high latency (we have the alert set to look at the last 100 samples, and notify us if 50 or more are over 500ms), that target is moved into our "High Latency!" summary screen. If the target falls back under that threshold, then it's then removed from the summary screen. This gives us a quick view into any of our targets that are currently experiencing issues.

## 5.2 Variable Substitution

A number of the alert parameters allow you to insert a variable which will be substituted for a value when the alert happens. Here is a list.

|                                          |                                                                   |
|------------------------------------------|-------------------------------------------------------------------|
| <b>{{Host}}</b>                          | Host Name & IP Address (no space, comma terminated)               |
| <b>{{Host.HostID}}</b>                   | Host Identification Number                                        |
| <b>{{Host.IPAddress}}</b>                | Host IP Address                                                   |
| <b>{{Host.HostName}}</b>                 | Hostname in DNS                                                   |
| <b>{{Host.AliasName}}</b>                | Custom hostname alias                                             |
| <b>{{Host.PreferredName}}</b>            | First non empty value for: alias, DNS hostname, IP address        |
| <b>{{Host.FullDisplayName}}</b>          | Preferred name followed by IP address in parenthesis              |
| <b>{{AlertName}}</b>                     | Name of Alert that fired                                          |
| <b>{{DestinationHost}}</b>               | Destination's Host Name & IP Address (no space, comma terminated) |
| <b>{{DestinationHost.HostID}}</b>        | Destination's Host Identification Number                          |
| <b>{{DestinationHost.IPAddress}}</b>     | Destination's host IP Address                                     |
| <b>{{DestinationHost.HostName}}</b>      | Destination's hostname in DNS                                     |
| <b>{{DestinationHost.AliasName}}</b>     | Destination's custom hostname alias                               |
| <b>{{DestinationHost.PreferredName}}</b> | Destination's first non empty value                               |

for: alias, DNS hostname, IP address

**{{DestinationHost.FullDisplayName}}**

Destination's preferred name  
followed by IP address in  
parenthesis

### Email Specific Variables

**{{attach 'timegraph' of \_last='60 seconds' width=700 height=200}}**

Timeline image for specified time  
period drawn to specified width and  
height

**{{attach 'tracegraph' of \_last='60 seconds' width=700 height=200}}**

Trace Grid and Latency Graph image  
for specified time period drawn to  
specified width and height

**{{attach 'screenshot' of \_last='60 seconds' width=700 height=200}}**

Screenshot image of for specified  
time period drawn to specified width  
and height (includes Timeline, Trace  
Grid, Latency Graph)

**{{attach 'pp2' of \_last='60 seconds' samples=100}}**

Export in PP2 format of data of for  
specified time period limited to the  
specified number of samples

## 5.3 Auto-Saving of Data

PingPlotter v5 saves data automatically. In V4, it didn't save automatically, and sometimes important data wasn't available when you needed it (unless you did some configuration that didn't always think the way you do). This is something that shouldn't happen to you in V5. In PingPlotter 5, it should "just work", saving what you want, discarding what you don't (if you show it what kind of stuff you don't want).

PingPlotter 5 writes all data in real-time to its local data store. This helps make sure that none of your data is lost at any point (for instance - if there is an unexpected restart, a power failure, or any other event that would put data data at risk).

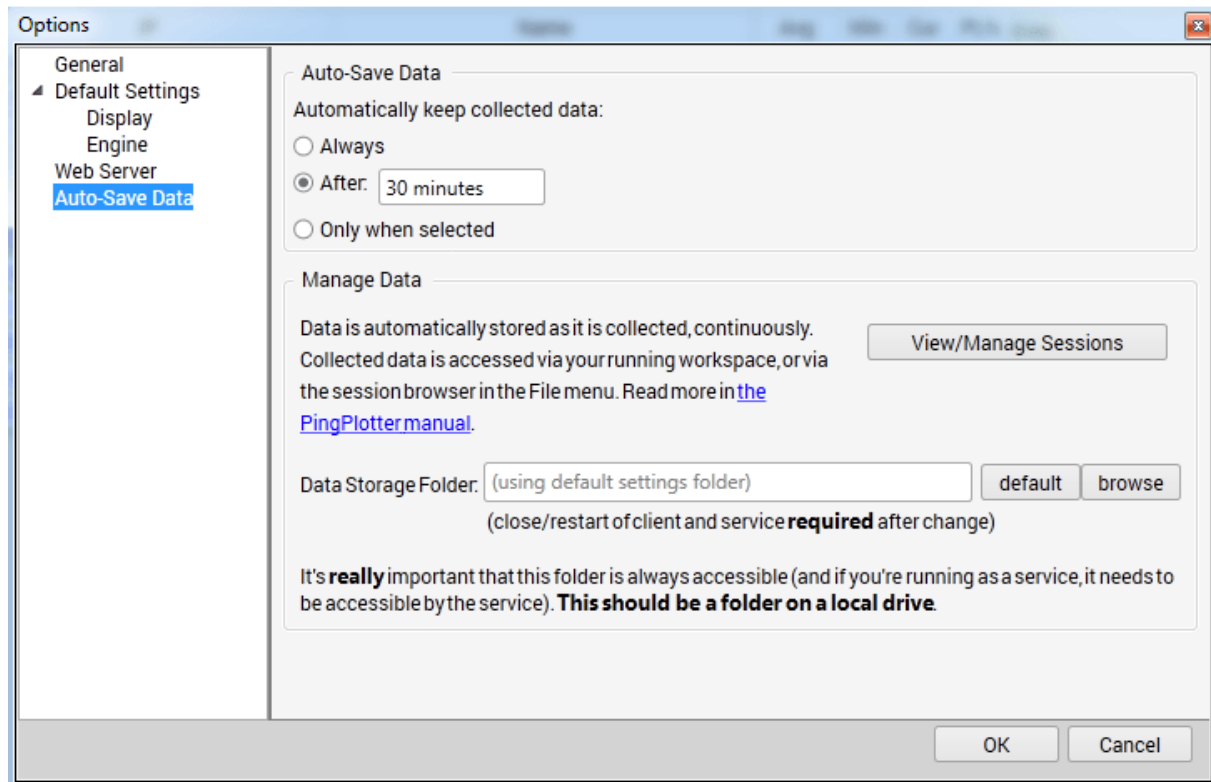
Long-term sessions (a session is a set of collected data from a trace to a target) are saved continuously,

and when you scroll back in history, the data is retrieved from disk. Closing and restarting will resume those sessions. Short term sessions are saved continuously, but if you don't collect data long enough, they are marked for deletion when closed.

When a session is marked for deletion, it will be fully deleted after you close and reopen PingPlotter five times. If you closed a session you really wanted to continue, you can reopen that session within 5 stop/restarts and continue. This behavior is a direct result of feedback from our users (and us) that prompting to save on close was frustrating.

## Configuring the definition of "short term" sessions

The auto save settings can be configured in PingPlotter by going to "Edit" -> "Options" -> "Auto-Save."



There are a few different options for how to recognize a session as one you want to keep around, or one you want to throw away when done.

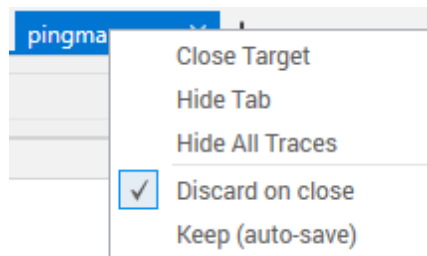
- "Always" - this will keep every target's data in PingPlotter's datastore. If you start tracing a target and stop after 3 samples, that "session" will get marked to keep.
- "After:" - this setting will only keep sessions that have met or exceeded the timeframe that is entered.

Use "5 minutes", or "1 hour" or similar.

- "Only when selected" - this will result in PingPlotter auto-saving data, but marking it to discard when you close. This is a great option if you have a handful of long-term targets that you always want to collect from, but you want to occasionally mix in some targets that you don't want to stick around.

## Keeping an individual session

If you're collecting data from a target and you decide you want to keep it (and you're in a mode where it's not yet marked to keep), you can switch to "Keep" mode (or discard mode) manually at any time. To do this, open a trace window (if it's not already open, use the "All Targets" summary to find it, then open from there), and right click on the target tab. Here we have options to "Discard on close," or "Keep (auto-save)."



The tab's "close" icon will show an "X" to indicate that this data will be discarded on close, and a checkmark to show that it will be kept.

## Finding a session

Once you've collected data but you've closed it, how do you reopen it (or delete it, or export it, or continue tracing)?

The previous trace sessions can be found in PingPlotter by going to "File" -> "Manage Sessions." From the session browser, we have the options to reopen or delete a previous session, as well as the option to export a session (which will save off a .pp2 file).

| Manage Sessions     |                        |                       |                       |         |                  |  |
|---------------------|------------------------|-----------------------|-----------------------|---------|------------------|--|
| Target Name         | Target IP              | Session Start         | Session End           | Samples | Persistence Type |  |
| ipv6:google.com     | 2607:f8b0:400e:c04::65 | 3/29/2016 1:37:29 PM  | 3/29/2016 1:37:51 PM  | 10      | Will Delete      |  |
| pingman.com         | 66.39.79.119           | 3/29/2016 1:36:28 PM  | 3/29/2016 1:37:53 PM  | 35      | Active           |  |
| pingplotter.com     | 216.92.151.75          | 3/29/2016 11:43:29 AM | 3/29/2016 1:37:53 PM  | 2534    | Active           |  |
| ipv4:www.google.com | 209.141.120.185        | 3/29/2016 10:48:36 AM | 3/29/2016 11:07:12 AM | 401     | Archived         |  |
| www.akamai.com      | 184.85.68.248          | 3/29/2016 10:48:32 AM | 3/29/2016 11:07:10 AM | 401     | Pause            |  |

There are four different persistence types in the session browser:

- Active - any trace sessions that are actively collecting data
- Pause - any trace sessions that are open in PingPlotter now, but are paused
- Archived - any inactive trace sessions that have been saved and are closed
- Will Delete - any inactive trace sessions that haven't been set to save. These will eventually be deleted.

All of the data in PingPlotter is saved to a file titled "sessions.ppdata" in one of two locations. If we're running PingPlotter as a Windows service the file's default location is in the C:\ProgramData\PingPlotter 5\ folder. If we're running the program as an application, then the file's default location is in our user application data folder C:\Users\\*\*username\*\*\AppData\Roaming where \*\*username\*\* is the username of the currently logged in user.

There is an option in the Auto-Save to change the default location of the sessions.ppdata file. If we specify a different location to store the file, we'll want to make sure to close and relaunch the program to get the changes to take effect. If we're running PingPlotter as a Windows service, then we'd need to restart the service (or restart the machine) to store the changes.

## Long term monitoring considerations

Now we know that data is being saved automatically, and you can open it, export it, and delete it from the session browser. Is there anything else to worry about?

Hopefully not!

You might be worried about configuring how much data to hold in memory (this was a common problem in PingPlotter 4 as more targets were added). One of the advantages of saving data automatically and

real-time is that we can count on it being there - and we don't need it in memory. PingPlotter automatically keeps a small amount of collected data in memory - enough for alerts to work, and your graphs to have accurate statistics. The rest it clears from memory and leaves on disk. If you scroll back in time, the data will automatically be pulled from disk for display and analysis. When you're done using data, it will be dropped from memory automatically.

There are a few things you can do to optimize memory and cpu use if you're tracing to a lot of targets (we'd call 250 "a lot", although that number is going up as we optimize more).

If you're tracing a long time to a lot of targets, you might:

- Hide target tabs for any targets you're not actively looking at. This minimizes statistics calculations and memory use.
- If you have a lot of targets with a lot of data used, don't scroll back a year with 200 targets on the all summaries tab. This will feel pretty slow as all 200 targets need to load from disk. When you need this, scroll back on a single target, or create a summary with just the ones you're interested in. We are continuing to improve this experience as PingPlotter gains capabilities.

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

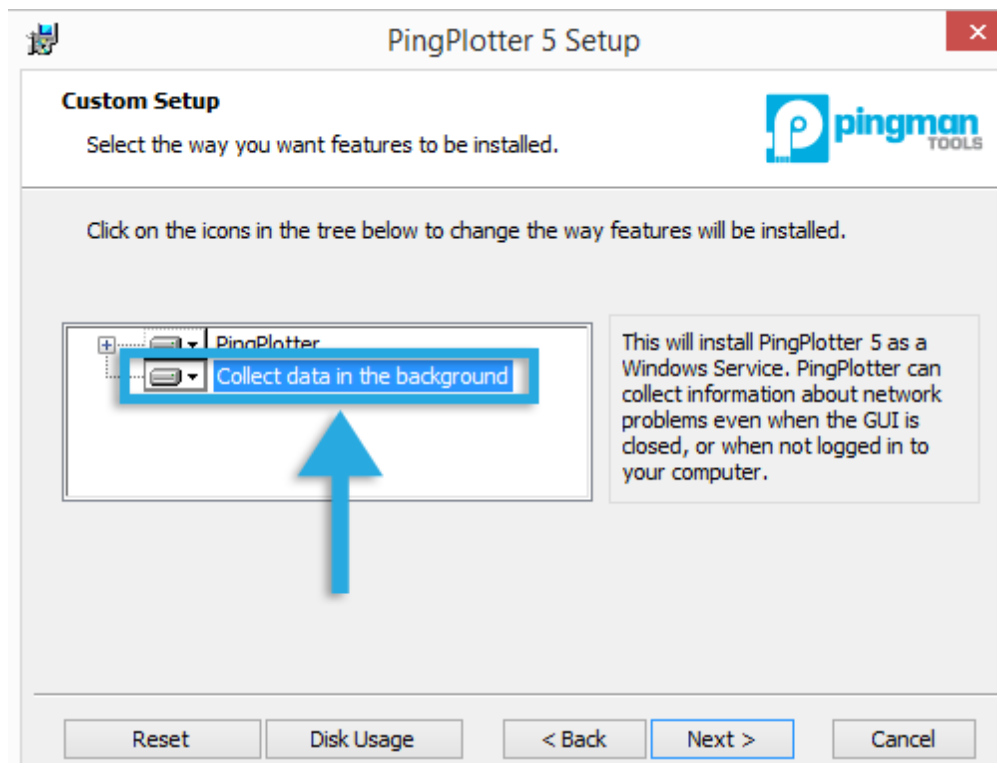
## 5.4 Running as a service

*Note: This documentation covers PingPlotter v5 and higher only. If you're using an older version, see the documentation that came with that version or upgrade to the latest version.*

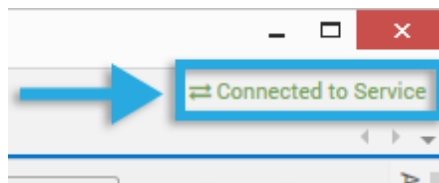
PingPlotter v5 can be installed as a Windows service - which allows the program to start running as soon as a machine is booted up, and continue running (even if a user logs out). This is a great option if we have a few targets that need to be monitored continuously, without interruption.

This option can be selected on installation (or if you've already installed - you can run the installer again and choose the option to "Modify"):





Once the installation is finished, the PingPlotter service will automatically start running. To access the service instance, we just need to launch PingPlotter, which will automatically connect to the service instance. If a new trace tab is opened, in the lower right hand corner of the program we'll see a status bar reading "Engine: local service" - letting us know that we're accessing the service instance of PingPlotter. If running as an application, this status bar will show "Engine: built-in."



## Benefits and Disadvantages

Running as a service in PingPlotter 5 is mostly upside. Good things:

- Data collection happens in the background, starting at boot. Logging out doesn't stop tracing.
- Some actions with security implications don't need to be configured - web server, Raw socket ICMP packet types being great examples.
- Super-easy, everything just works.

Disadvantages:

- Not many, but the security implications have a negative, too. Some alerts, like "launch an executable", could be used for evil when running as a service.
- Communication uses TCP port 9636, with no option to change (currently).

## 5.5 Handling Multiple Targets

PingPlotter Standard can trace up to 2 targets simultaneously in the same instance.

PingPlotter Pro can be used to trace to multiple different targets within the same instance (and each target's configuration can be independently managed, too!). There's no "hard limit" on the amount of targets you can add in PingPlotter Pro - but we've successfully tested up to 250 here at Pingman HQ.

### Adding new targets

There are a few ways to add new targets to PingPlotter. If you've just opened up the program for the first time, odds are you're on a "(new trace)" tab - and the "Target Name" field is empty (with a blinking cursor in it):

All you need to do is (if you haven't guessed it by now) enter your target's DNS name or IP address in the empty field, and hit the big green "Start" button on the left hand side (the "enter" key on your keyboard works here as well).

Wondering how to get another "(new trace)" window open? You can use the "File" -> "New Target" option, or, in PingPlotter Pro, just click on the "+" symbol in the upper left hand corner of the program:

If you're on a summary screen in PingPlotter, you can quickly add targets by entering them in the "Target name" field, and pressing the start button (or the enter key on your keyboard). Targets added via this method will automatically start tracing and are added to your list of all targets (and if you're on a custom summary screen, they'll be added to that summary as well - we cover this in more detail in the summary graphs section). A tab *won't* be opened if you're using this method - but it's a good way to quickly add targets and start tracing to them.

### Loading a list of targets

If you've got a list of IP addresses, DNS names, or both, you can "bulk add" them into PingPlotter Pro as well. You'll want to make sure your list is formatted correctly to accomplish this. In a text editor (notepad works just fine here) create a list of your addresses - one target per line. Once you've got everything added in, save the file with a "\*.lst" extension.

Once you've got your list saved, in PingPlotter go to "File" -> "Load Target List File," navigate to your list

file and open it - and PingPlotter will load up the targets from the list.

Voila! PingPlotter will start to load your list of targets, one at a time - exactly as though each target was entered into the “New Target” field. Any invalid targets will prompt an error - which will prompt \*after\* PingPlotter gets through your entire list.

## Selecting Multiple Targets

You can select multiple targets and apply changes to them (pause, change the trace interval, use a different configuration, or add them to a summary screen). This is a pretty easy process, too. Hold down the “CTRL” key on your keyboard, and click on a few different targets. If the targets you’re needing to select are in order, you can click one, hold down the shift key, and click on the last one in the list to get the whole group.

Once you’ve got multiple targets selected, any changes made in the [target bar](#) will be automatically applied to the selected targets. There are also a few options (pause, resume) available via the right click menu.

## Docking, Floating, etc

By default, each target in PingPlotter Pro is shown in the “All targets” summary screen (and doesn’t have it’s own tab unless you open one for it). This works, but may not be exactly what you’re after. Maybe you want a few targets side by side. Or maybe you have 8 monitors and want 64 targets up filling your high def “war room screen” (and if this is the case... we salute you). There is definitely a way to achieve this!

Once you open a target (or summary) into a tab, that target is living in its own “dock” control. If you grab the tab and drag it away from PingPlotter, you can position it somewhere within the PingPlotter widow - or drag it out into its own “stand-alone” display.

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

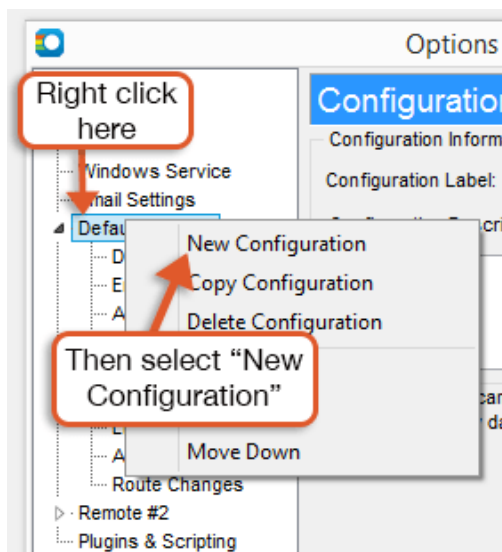
## 5.6 Named Configurations

So, we know that PingPlotter Pro allows us to trace to multiple targets in the same instance. What if, however, we want to use different settings on different targets? For instance - for one target we want to use TCP packets, but want to use ICMP packets on another. Or maybe you want to trace a few of your targets from a remote agent location.

The good news is PingPlotter Pro can do this through the use of “Named Configurations.” The bad news?

There is no bad news - this is all very good stuff.

We can access (as well as create and manipulate) our named configurations in PingPlotter by going to the “Options” menu (“Edit” -> “Options”). When you first install PingPlotter you’ll have one named configuration (appropriately named “Default Settings”). Under the configuration name we see a number of other “configuration specific” areas (such as “Display” and “Engine,”). The values that we configure in these areas are all a part of this specific named configuration. At any time, we can edit the name, or make changes to any of the values.



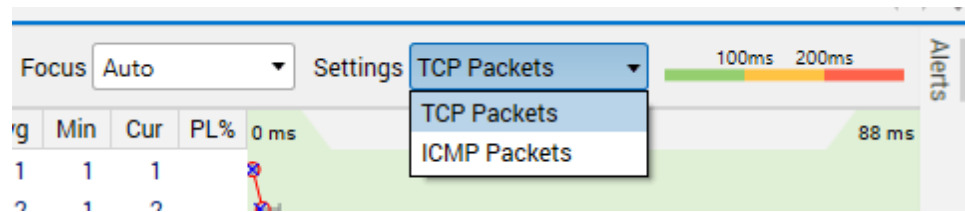
To create a new named configuration, all we need to do is right-click on any existing configuration, and then select “New Configuration.” If you’ve got a configuration already set up, and need to set up another that’s similar (with minor tweaks), you can right click on that configuration and select the “Copy Configuration” option. Any changes that you make to a specific configuration are *\*not\** carried over to any other named configurations.

Once we’ve created a new configuration, it’s always a good practice to rename it (and give it a good, meaningful name). Some example configuration names could be:

- TCP Port 80
- Auto-Save every 30 minutes
- VoIP
- Remote via our website

There’s also an option to enter a description - which can prove helpful (in case you come back to a configuration and wonder why it was made, and what it was for).

Once you’ve got everything set up to your liking, and close down the “Options” menu, you’ll notice a drop-down box in the target bar labeled “Settings.” This field lists all of the named configurations we have set up.



Changing this drop-down will change the currently running target to use that named configuration. You can change multiple targets to use a specific configuration by selecting them on the summary screen, and changing the settings field (the changes will take effect immediately once selected).

*\*\*The feature in this topic is exclusive to PingPlotter Pro. See our [product comparison](#) page for more details\*\**

## 5.7 Reporting

PingPlotter doesn't have any \*printing\* options (who needs paper anyways?), but it does have a variety of options available to help output data (which will let you manipulate it in your favorite software package from there).

There are several methods here:

### The graphs

A PingPlotter image (the combination of graphs you see on a target screen) should be your go-to first report. These images have been used by many, many thousands of people to communicate their problems to their provider.

If the image you see doesn't seem compelling, or doesn't capture the right picture, you may need to adjust your view. You can drag the time graph back, and focus is on a period that shows the problem. Depending on the problem, you might want to widen your view a bit to show periods of OK along with the periods of problem. Maybe 12 hours if your problem period is 6 hours, or 2 hours if it's 45 minutes. Right-click on the time graph to pick the appropriate time, then drag back and forth.

Once you have the time graph focused on the period you want, turn on the appropriate hop graphs. Hops that have latency or packet loss that isn't represented in the final destination (a topic we cover in more detail [here](#)) shouldn't be highlighted, while ones near the origin should be.

Once you've got that view, you can get the upper statistics to show the part you find important. You can leave it at "auto", which will have the upper and lower graph show the same time frame. Or you can switch the upper graph to a time period less than the lower graphs. If your lower graph is focused on 12 hours, you can set the upper one to 6 hours, and then double-click on the time graph to move the focus

period. This will let you pick the statistics that show your problem and will also show the lead-in and exit to the problem.

Now span a picture via Edit -> Copy as image or via File -> Save as Image

This concept is more succinctly covered in our [video tutorial](#).

There are ways to export, too, but the analysis tools coupled with the image creation tool really create a powerful case.

There are a few ways get a picture of your PingPlotter graph(s). The quick way is to select the "Edit" -> "Copy as image option" (which will copy the graph you're viewing to your clipboard). The column and graph sizing will match up exactly to what you see on your screen - so make sure everything you want to show is explained. Another method is the "File" -> "Save Image..." option.

## Statistical text export (inspired by command line traceroute)

In PingPlotter, you can choose the "Edit" -> "Copy as Text" option to copy the raw data to your clipboard. This is copied in a format that's similar to \*most\* text-based trace route programs. It's important to note: the "Focus" option on the main screen is used to decide how big of a period to include in this.

## Comma delimited text file

This option is built to import into a program like Microsoft Excel - so you can manipulate the numbers and create output in different formats. To export a comma delimited text file, use the "File" -> "Export to Text File" option. From there, you'll be able to specify the file you want to export (and a couple of other options). You can either export all samples in memory, or the range as specified on the main screen.

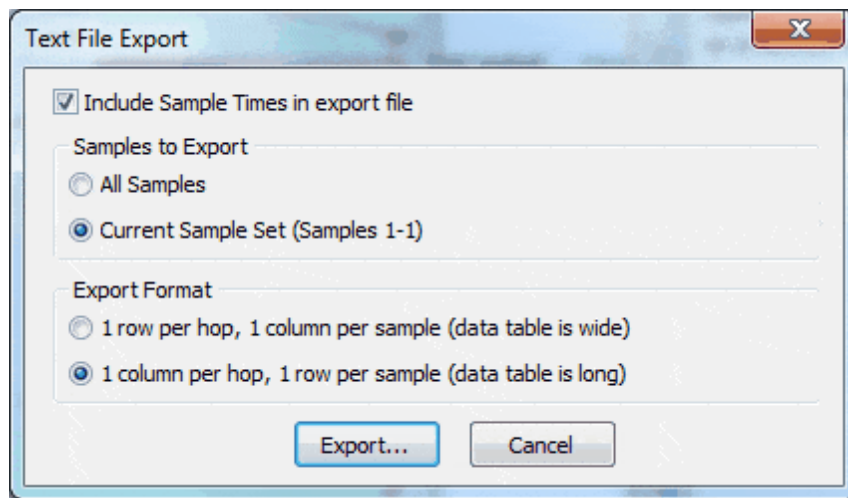
The "Include sample times in export file" option will specify whether or not to include the time each sample was taken at. If you don't have this turned on, all of the samples will be output, but you won't get corresponding times. Enable this option to include the times.

We discuss more reporting options [in our knowledge base](#).

## 5.8 Exporting data for further analysis

If you're looking to export data from PingPlotter, this can done via the "File" -> "Export to text file..." menu option. This option allows you to take all of the data currently in memory in PingPlotter and save it to a text file which can then be loaded into Excel (or a similar tool).

We're going to briefly discuss the options here. If you want more information and an example Microsoft Excel spreadsheet, check our [knowledge base](#) (in particular, [this topic](#) and [this topic](#)).



### Include Sample Times in export file

If left unchecked, the times for each trace don't get saved - only the data.

### Samples to Export

Select **All Samples** if you want to export your whole trace to text. Select **Current Sample Set** if all you want to save is the currently displayed sample set shown on the Time Line graph. The current sample set is the setting you'd use if you're wanting to email trace data to an ISP, etc. - though just saving a graph would be a better option.

### Export Format

PingPlotter gives you two different export formats to save your data in. Both are shown below (Note: if you're using Excel, the second "1 column per hop" usually works best).

## Examples:

### One row per hop, one column per sample

```
,Sample Times,,9/11/2014 11:13:29 PM,9/11/2014 11:13:30 PM,
 9/11/2014 11:13:31 PM,9/11/2014 11:13:32 PM,9/11/2014 11:13:33 PM,
 9/11/2014 11:13:34 PM,9/11/2014 11:13:35 PM

1,,,N/A,N/A,N/A,N/A,N/A,N/A,N/A
2,loop1.bois-dsl-gw1.bois.uswest.net,216.161.136.254,10,10,10,10,10,20,10
3,100.fa2-0.bois-agw1.bois.uswest.net,207.108.229.29,30,20,10,10,10,10,10
4, ----- ,207.108.224.247,20,10,20,10,20,20,10
(.... middle data snipped for brevity)
16,192.ATM7-0.GW5.SJC1.ALTER.NET,152.63.54.21,60,51,50,50,50,50,50
17,digexoc12-gw.customer.alter.net,157.130.214.154,50,50,50,50,50,50,50
18,gsr-01-p2-0-a00a02.af.sjc5.digex.com,164.109.130.26,50,50,50,51,50,50,50
19,lc1.com,164.109.154.154,50,50,50,51,50,50,50
```

### One column per hop, one row per sample

#### Host Information

```
1,,
2,loop1.bois-dsl-gw1.bois.uswest.net,216.161.136.254
3,100.fa2-0.bois-agw1.bois.uswest.net,207.108.229.29
4,-----,207.108.224.247
(... middle hosts snipped for brevity ...)
16,192.ATM7-0.GW5.SJC1.ALTER.NET,152.63.54.21
17,digexoc12-gw.customer.alter.net,157.130.214.154
18,gsr-01-p2-0-a00a02.af.sjc5.digex.com,164.109.130.26
19,lc1.com,164.109.154.154
```

#### Sample Information

```
"9/11/2014 11:13:29 PM",N/A,10,30,20,30 .. snipped .. 60,50,50,50
"9/11/2014 11:13:30 PM",N/A,10,20,10,30 .. snipped .. 51,50,50,50
"9/11/2014 11:13:31 PM",N/A,10,10,20,30 .. snipped .. 50,50,50,50
"9/11/2014 11:13:32 PM",N/A,10,10,10,30 .. snipped .. 50,50,51,51
"9/11/2014 11:13:33 PM",N/A,10,10,20,30 .. snipped .. 50,50,50,50
"9/11/2014 11:13:34 PM",N/A,20,10,20,30 .. snipped .. 50,50,50,50
"9/11/2014 11:13:35 PM",N/A,10,10,10,30 .. snipped .. 50,50,50,50
```

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

## 5.9 Route Changes

In PingPlotter v5, route changes can be viewed through the use of the program's dynamic focus period - and any route changes that may be present can be viewed by navigating through the time graph for a target.

To get started, set the "Focus" field to "Auto":



As the time graph is moved forward/backward in time, the upper trace graph will automatically be updated to reflect the routes that were in use during that time period.

There are a few tell-tale signs to keep any eye out for when trying to spot route changes. The first (and most obvious) would be any length changes in the route (if it shrinks or expands by a few hops) as you scroll. Another great way to spot route changes is to keep an eye on the "Count" column in the upper graph. If one (or more) hops is displaying a count that's off from most of the other hops, it's a good indicator that specific hop was recently added (or re-added) to the route:





| Hop | Count |
|-----|-------|
| 1   | 180   |
| 2   | 180   |
|     | 180   |
| 4   | 180   |
| 5   | 22    |
| 6   | 180   |

You can drill down even further by double clicking in a time graph to bring up a [focus area](#) to see the route that was in use for a more specific time period.

# Part

---



VI

## Understanding Output

## 6 Understanding Output

### 6.1 Discovering a route between you and a target

Let's run through a basic scenario where we discover the route between your computer and an interesting destination (maybe one you're having a problem with).

Normally, when troubleshooting a problem, you want to run PingPlotter against the server where you're experiencing problems. Maybe that's a web server (in which case, you'll want to enter that web server's address); maybe it's a game server (in which case you'll want to enter that game server's address). If you aren't experiencing problems with your network connection or something you access with it currently, no worries! Just try and think of something you access regularly with your Internet connection (such as [www.PingPlotter.com](http://www.PingPlotter.com). or [www.Google.com](http://www.Google.com)) to use in the exercise below.

**We're assuming here that you've downloaded and installed PingPlotter. If you haven't, please see the [Downloading and Installing PingPlotter](#) section for instructions on how to do so.**

So load up PingPlotter, and let's get started!

1. Enter the IP Address (i.e. 129.41.62.29) or the DNS Name (i.e. [www.PingPlotter.com](http://www.PingPlotter.com)) of a destination you may be having problems with into PingPlotter's Address to Trace: input box. Note: we just want the name of the destination. You would not enter <http://www.problemserver.com/index.html> here. What we want is between the "http://" and the "/index.html."

For now let's leave all the other settings you see on the screen as they are.

2. After you've typed in the address, either press the "[Start](#)" button, or press the "enter" key on your keyboard. The trace then starts, and you'll see the upper Trace Graph populate with the route information to the target you entered. The Timeline Graph for that target will be displayed also below the Trace Graph.

One thing that is kind of an 'ah ha!' moment for a lot of first time PingPlotter users is seeing that you really do have that many devices your network traffic passes through to get to web sites, servers, etc. If you click on a web page link, that 'click' is passed on by all those hops to that final web server/page, that web server executes that click, passes the information back to you through all those hops and you see it on your browser.

It's important to cover the concept of a "[Sample Set](#)," because we're going to be mentioning it a lot. The sample set is defined by the "[Focus Time](#)" value (which is right under the "Address to Trace" and "Trace Interval" fields). The "PL%" and "Avg" columns in your trace graph are all computed off of this number. If this value is set to 10, PingPlotter uses the last ten samples it's done and bases everything off

that number. If it's set to 20, PingPlotter uses the last twenty traces it's done, etc. As we go through what you're actually seeing on the graphs, just remember that the "Sample Set" is based off the value in the "Focus Time" value, and that number represents the number of samples - starting with the most recent and working backwards.

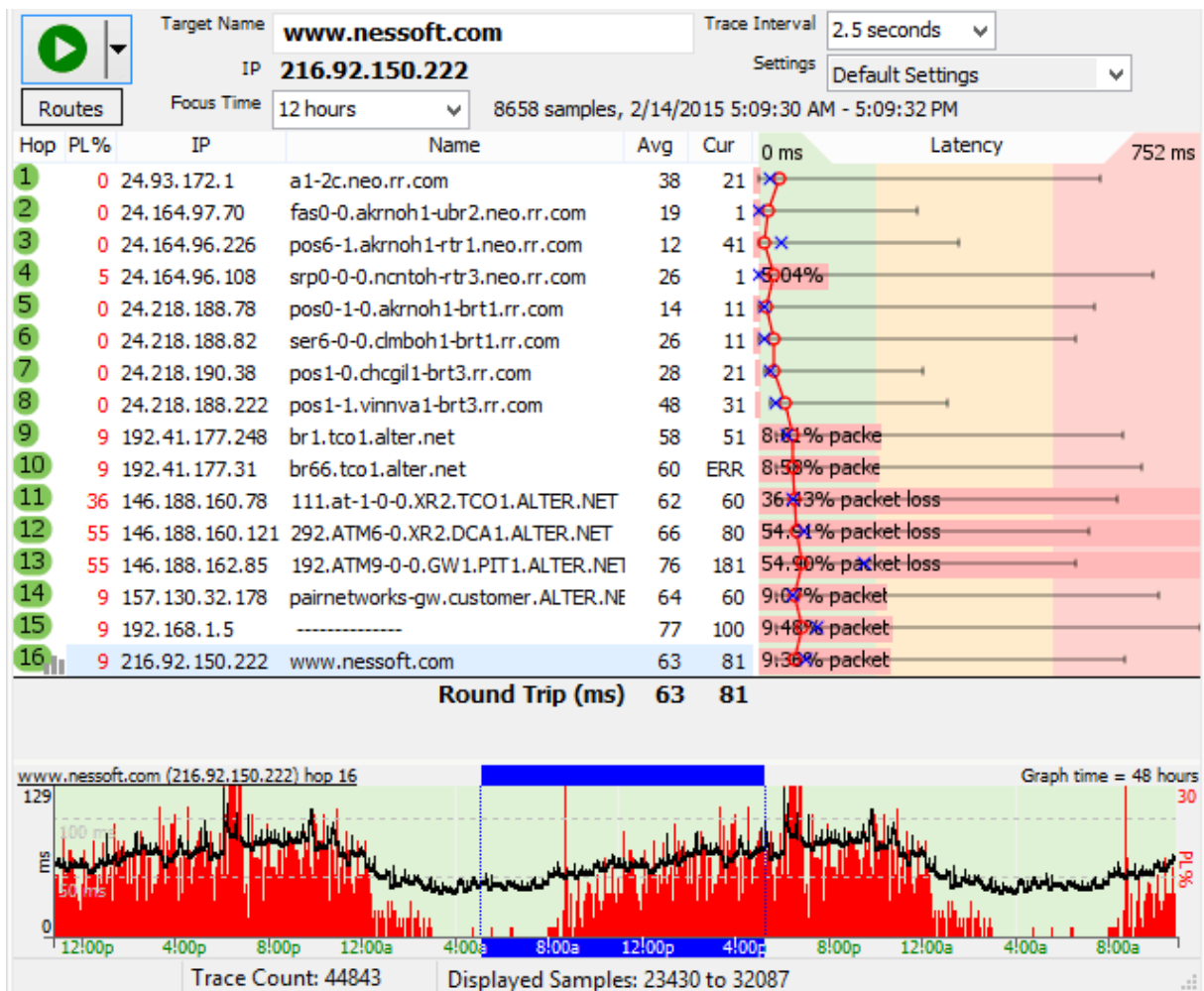
## Some things to consider before we move on:

- We cover what's "normal" for latency and packet loss in a [knowledge base](http://www.pingman.com/kb/42) entry at <http://www.pingman.com/kb/42>
- If you get a "Destination Address Unreachable" message at the last hop in the trace graph, this means something between your computer and the final destination isn't receiving and/or returning packets. We cover this in detail in the PingPlotter section of [our knowledge base](http://www.pingman.com/kb/8) at <http://www.pingman.com/kb/8>. If some hops are responding, you might try using a different target address (i.e.: try [www.pingman.com](http://www.pingman.com) instead of the address you entered the first time).
- If the final destination is working (i.e.: the Round Trip row is showing), but some of the earlier hops are not, then don't despair! This could be normal. See knowledge base articles <http://www.pingman.com/kb/24> and <http://www.pingman.com/kb/29> for possible causes of this behavior.
- If you're entering an IP address and would like to "label" that address, or give it a "friendly name" to make it easier to find in history later, you can do that by entering the IP Address, then a space, and then the label. For instance, if you have a core router that you'd like to see displayed in PingPlotter as "Core Router" (minus the quotes), you'd enter it as "192.168.0.1 Core router" (omitting the quotes and substituting the 192.168.0.1 for the IP Address for the actual IP Address for your router). For more details on this, see <http://www.pingman.com/kb/23>.

## 6.2 Interpreting Results - A Quick Example

So let's get into some specific examples of how to interpret the results from PingPlotter.

For the first example, you're getting intermittent packet loss to [www.nessoft.com](http://www.nessoft.com). What can we determine from the graph below?



First off, the final destination (hop 16) shows 9% packet loss. There's a problem somewhere in the route, but we need to determine where....

Hop 4 shows 5% packet loss. Hop 5 doesn't show packet loss, though, so you know that the problem in hop 16 isn't because of hop 4. Hop 4 is likely just a router using a different CPU path for TTL=0 packets than it does for routing data through.

Hop 9, however, shows 9% packet loss, and this packet loss is carried on through to the final destination. This is a **huge** indication of where the problem lies.

Now, all we know from this is that the problem happens after hop 8. We don't know if it actually happens because of CPU overloading in hop 9, a router problem in hop 9 (or even on the exit side of hop 8), or if it's the connection between hop 8 and 9. A little bit more troubleshooting is needed for this.

Digging deeper, we can see (from the domain names) that hop 8 is in the rr.com domain, while hop 9 is in the alter.net domain. Also, the IP addresses show decidedly different ranges. This is a strong clue that

it's actually the connection between hop 8 and 9 that's causing the problem. It's likely that there's not enough bandwidth between those two locations.

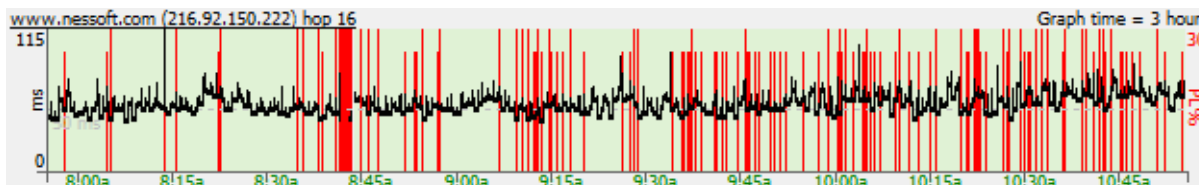
## 6.3 Finding the source of the problem

We learned how to do a basic PingPlotter trace in the [previous exercise](#). What we want to do now is take a pre-prepared save file that contains about 2 ½ days worth of fictional data and do more hands-on work with PingPlotter.

First off, you need to get the data file downloaded for this exercise. The file you need to download is at [www.pingplotter.com/gsg/www.nessoft.com.pp2](http://www.pingplotter.com/gsg/www.nessoft.com.pp2). Save this file to your desktop so it will be easy to find (of course you can select another location - just remember what you specified).

Load up the save file in PingPlotter by going to "File" -> "Load Sample Set," and then browse to your desktop (or wherever you saved the nessoft.pp2 file) and select the www.nessoft.com.pp2 file, then click the "Open" button. Ta da! Two and a half days worth of data for us to play with.

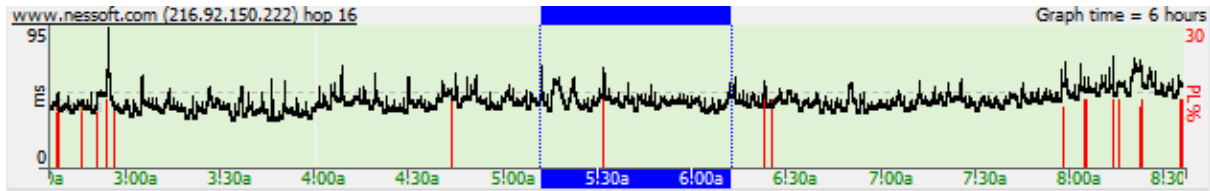
### Let's go over some concepts concerning the Timeline Graph.



1. Let's start out with this concept – red on a graph is bad. All those red lines you see on the Timeline Graph indicates Packet Loss for that particular time period. Remember, that means that PingPlotter didn't get an answer back. There are times where you could have a flaky router, or even a server that is de-prioritizing ICMP packets. If you'll remember from the [How PingPlotter Works](#) section, this is how PingPlotter gets its information. You could have a perfectly fine connection through that router, but PingPlotter will show it at 100% packet loss. For the most part though, when you see red on the Timeline Graph it means PingPlotter wasn't able to get to that server.
2. The black line on the Timeline Graph is average latency for the time period you're looking at. When you zoom in on the graph throughout the next two steps you'll be able to see the individual pixel-wide points on the graph.
3. The default Timeline Graph scale is 10 minutes - but what if we want to see more? Right-Click with your mouse button on the Timeline Graph, and you'll see that you can show anywhere from 60 seconds on up to 48 hours worth of data within the Timeline Graph. Go ahead and select 6 hours. Notice that you're now looking at six hours worth of trace data.

Feel free to select different time periods and see how the contents of the Timeline Graph reflect what you select. What we want to illustrate here is that via the Timeline Graph's right-click mouse menu, you can easily zoom out to look at a trend for a day, and then zoom in to a spot that looks interesting – down to 60 seconds worth of data.

4. The "focus area" on the Timeline Graph (which is brought up by double clicking anywhere on a timeline graph) shows you the current sample set that you're viewing.



Remember that value is set in the "Focus Time" value. When you first load up this save file, PingPlotter is starting at the last trace done before the data was saved, going back 10 traces, and then calculating the values for the Trace Graph based on that value. Note also that PingPlotter shows you the date and time information for the sample set that you're looking at in the area up above the Trace Graph.

You can change the focus of the Timeline graph, and subsequently the data in the Trace Graph, by double-clicking on any area of the Timeline Graph you want to look at. In case you're wondering, you can have an active trace going while you move around the Timeline Graph. You do not need to save the data, stop the trace or do anything else for that matter. Just double-click on the area you want to see.

Now change the Focus Time to 150 samples. See how the values in the Trace Graph change to reflect those additional 150 samples/traces? Change the Focus Time to ALL, or 0 (0=ALL in this context).

Whoa! If you look at the Sample Set Time above the Trace Graph you'll now see that you're looking at a lot of data. In fact, you're looking at all 44,843 traces, or ALL of the traces that were in the file we loaded. Your PL% and Avg columns in the Trace Graph now reflect all those traces. Now you can change it back to something smaller (100 is generally a good value to use).

5. If you're not there already, right-click on the Timeline Graph and set the time you want to see to 6 hours. To move back through the data you click-drag, or left-click, and while the left button on your mouse is held down you move the graph to the right. To move forward in time, you click-drag to the left. Are things starting to "click," as far as how you can move back and forth through time on the Timeline Graph? If you have a lot of samples to move through, do yourself a favor and set the graph scale accordingly (again, by right-clicking and choosing a time) before you do. For instance, if you've zoomed in to where you're looking at 5 minutes worth of data and then need to go back twelve hours – set your graph scale to something like 3 or 6 hours before you start.

Tip: You can also use the keyboard to navigate backwards/forwards, as well as quickly get to the

beginning or end of Timeline Graphs. The shortcut keys used to do this are covered in the [Interface Graphs](#) portion of this manual.

Hopefully by now you've been clicking all over the Timeline Graph, and seeing that the Timeline Graph does indeed change when you select a new sample set to look at. So how do you get back to the current time/date? Bring up the Timeline Graph's right-click menu and select "Reset Focus to Current." If you don't see that listed in your right-click menu....well, then, you're already current.

At this point if you're thinking, "Hmmm, this Timeline Graph stuff is cool, but I want to see a Timeline Graph for Hop 1 too!", then today is your lucky day. You can do this by double-clicking on that (or any) hop - or by right-clicking and selecting "Show this Timeline Graph." Double-clicking again will hide that same graph. If you have a lot of Timeline Graphs open, clicking once on the hop for that Timeline Graph will cause that Timeline Graph to flash briefly so you can pick it out of the list.

## Checkpoint

So let's summarize what we've done, and what we know at this point about the Timeline Graph.

- You can change the scale by right-clicking and selecting one of the values listed.
- The focus area on the Timeline Graph shows you the current sample set, and you can change it easily to reflect the time period you want included in the Trace Graph.
- You can move around within the Timeline Graph by click-dragging or using the shortcut keys.
- By changing the graph scale, click-dragging then changing the scale back, you can zoom in and out of different time periods you want to analyze.
- PingPlotter tries to keep your "focus period" in view on the timeline graph, and you can use this if you're zooming in on a period. Focus the period you're interested in (double click on it), and then zoom in.
- If you want to get up to the very last sample set in a save file, or the last sample done if PingPlotter is actively tracing a target site for you, "Reset Focus to Current" on the right-click menu for the Timeline Graph will get you there.
- If you want to turn on or off visibility for a Timeline Graph for a hop, you can do so by double-clicking that hop, or by right-clicking on that hop and selecting "Show this Timeline Graph".
- To quickly find a Timeline Graph for a particular hop, if you have a lot of Timeline Graphs showing for instance, click once on that hop and the associated Timeline Graph will flash.



## 6.4 Finding the source of the problem - part 2

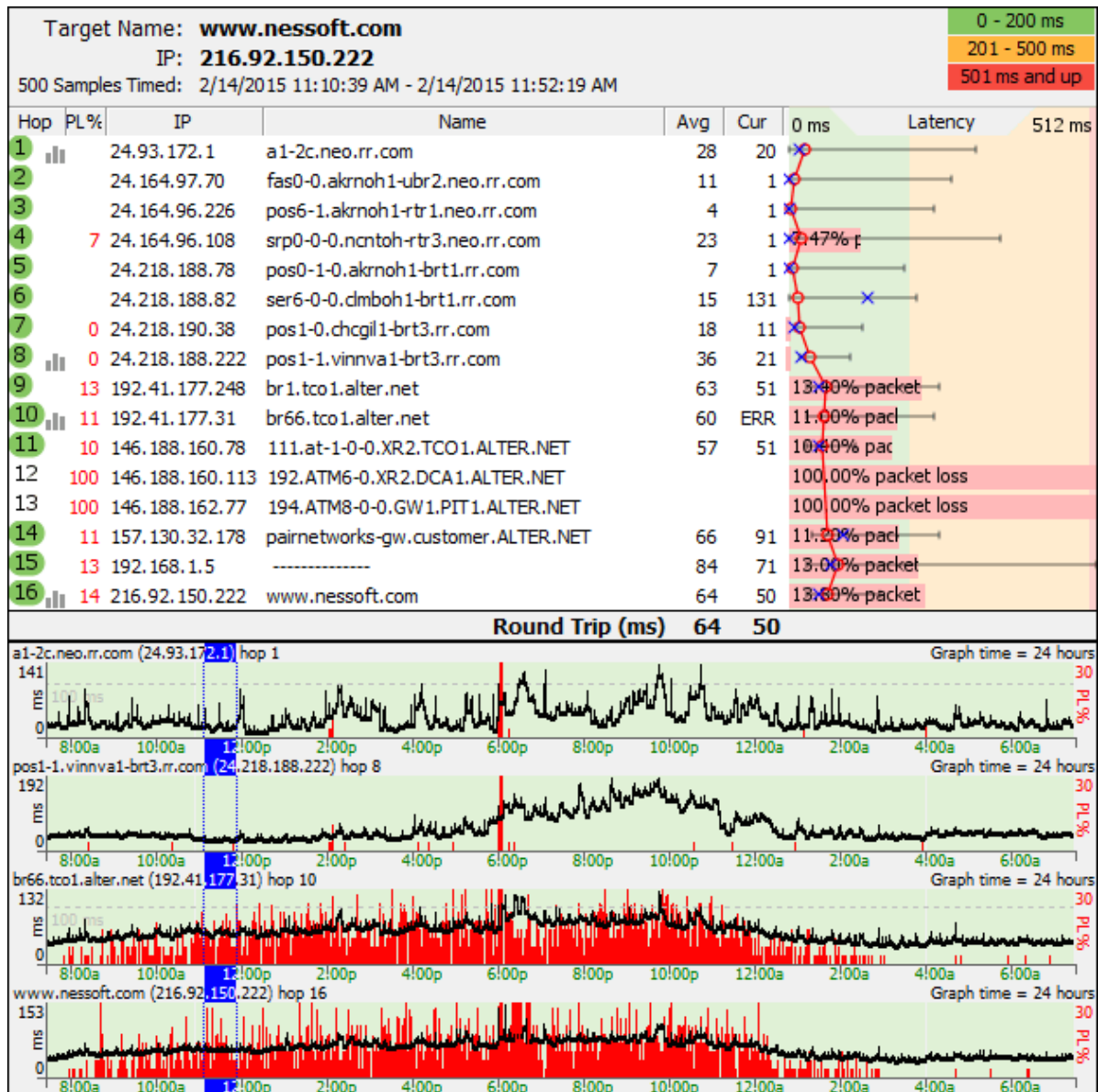
### Let's apply and build on what we've learned and dig deeper.

The black line on the Timeline Graph shows you the latency over time. As you zoomed in on the graph earlier you saw that really what you're seeing is pixel-wide entries that represent the round trip time, or latency for the host/device who's Timeline Graph you're looking at.

Red, on the other hand, represents packet loss (PingPlotter didn't get anything back). We have numerous articles that go over different reasons why you could see packet loss in the Knowledge Base at [www.pingman.com/kb](http://www.pingman.com/kb). Our goal here is to help you understand at a basic level how PingPlotter can help you figure out why that loss is happening.

Packet loss, of course, isn't the only thing that can cause poor performance for your particular application. The other big factor is latency. For online games, for instance, latency is a killer. Let's review our sample data, and look at a trouble spot so we can learn more how to interpret what PingPlotter gives us that no other tool can.

Let's look at the time period between 11am and 12pm on February 14, 2015. Let's take the approach that we were browsing the web at that time and were running pcAnywhere into our computer at work. The pcAnywhere session goes to pot.



1. Set "Focus time" to somewhere between 30-60 minutes (we're using 60 minutes here because it's easy to see the focus period on the 24 hour graph).
2. Change the Timeline Graph scale to 24 hours (remember: right-click then pick 24 hours), and then scroll all the way to the beginning of the graph
3. Take a look around at that time between 11:00 am and 12:00 pm. Double-click a time period somewhere around the 12:00 pm time. You'll notice the focus lines appear on the time graph, and the upper graph will show packet loss.
4. Wow, look at that packet loss! 13% at hop 9, 14% at hop 16. Notice how the packet loss is added at hop 9, and then all the downstream hops also show high packet loss. This is a strong (compelling, in

this case) indicator that hop 9 (or the link between hop 8 and hop 9) is the reason we're seeing packet loss.

5. Notice the packet loss trending over time. This indicates some kind of time-based load problem. Also, the fact that the packet loss starts at the junction point between rr.com and alter.net indicates a possible problem at the connection between these two providers. It's possible that rr.com doesn't have enough bandwidth to service needs.

6. Let's turn on a couple more timeline graphs. Double-click on hop 9 and hop 8. Notice the difference in packet loss. Notice, too, that the latency is still relatively high, even at hop 8, and it's also showing time-based problems. Double-click on hop 1, and notice that that \*to\* also shows a time-based latency problem. This is a separate problem from what we're seeing at the hop 8 to hop 9 junction point! Looks like we should contact rr.com and find out if they can help us solve this problem!

7. Try zooming in on a problem-period by changing the timeline graph scale to something lower. When you see an interesting period, double-click on it to "focus" it, and then change the timeline graph scale to zoom in or out. This gives you a close-up view of the data with more detail.

Now, it should be noted that spikes on your graph may not be really high latency. A spike in the graph is high latency for that period displayed on the graph. The Timeline Graph auto-scales itself, so to see what that high spike is you need to look to the left-most part of the Timeline Graph. In this case the value we're looking for is 87ms.

This was a really quick example of moving around within PingPlotter and digging out information about not only what's going on with a connection, but really focusing in on a problem. Let's move on now and talk about collecting this data over time and preparing a good case for your ISP (or someone able to solve this problem!).

## 6.5 Interpreting Results - Longterm Monitoring

PingPlotter allows you to use the timeline graphs to zoom in on any particular time, so even if you weren't there (or didn't save an image) when something was happening, you can still recover that exact image later. You shouldn't ever have to be sitting in front of your computer when an outage happens, or you experience other problems on your network, to get the data you need from PingPlotter.

### The Scenario:

You're having outages (or situations you want to communicate to your ISP) randomly throughout the day (let's say twice a day). The problem is that you can't be there every time an outage happens so you can [save a graph image](#) .

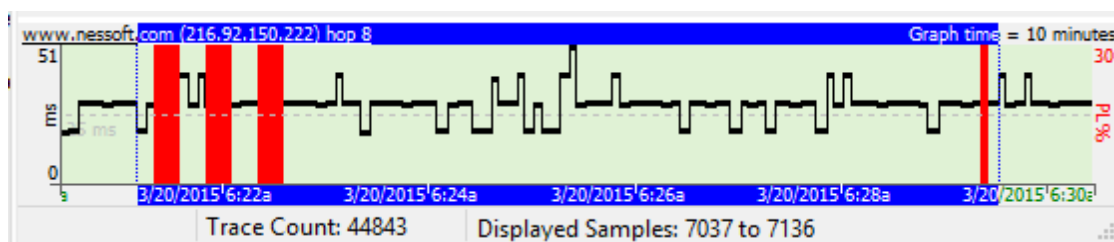
For this example, you're keeping 24 hours of data in memory or even more. We suggest that you normally use 2.5 second trace intervals and keep 200,000 samples in memory (this is almost a week's worth of data). You can change the number of samples to keep in memory in the [Auto-Save](#) section under the [Edit/Options menu selection](#).

Using PingPlotter's [timeline graph](#), you can see over the past X time period (see below) to identify a time period where there was problems. Problems will demonstrate themselves as packet loss (red), or high latency.

Now you want to show the route, and the packet loss/latency in the upper graph for that time period. Since that time has already passed, you need to change the focus of the upper graph to that time in the past.

First off, you need to make sure your "Focus Time" focuses in on just the period in question, so let's change that to 100 (it's important to **not** have it set to 0/all, but to have it be a number smaller than the number of samples in memory to be able to focus the upper graph). Right click on the timeline graph and pick a reasonable period of time to set the viewable time period. For instance, you may want to set it to six hours so you're not scrolling forever. You can then "click and drag" the graph to the left to go back to the time period you want to focus on and drag it to the right to go forward in time.

Double-click on the "problem period" in the lower graph. You'll see a focus rectangle appear on the lower time graph, and the upper graph will change to represent the data you have "focused" in the lower graph. Once you've done this, you might want to change the scale of the lower graph to show more detail. Right-click on the lower time graph again and change the scale to an hour (or maybe even 30 or 10 minutes depending on how long the outage was). The focus rectangle should still be visible. You can fine tune the data being displayed in the upper graph by double-clicking on the lower graph again.



Using these techniques, you should be able to zoom in on exactly the right data to best illustrate the problems you're seeing. You can look at the data after the problem occurs and get the perfect picture and not have to sit there watching PingPlotter all day and night.

You can auto-save that data by for instance having the [auto-save](#) function in PingPlotter create new files every day, and then load up a prior day to do the same thing you did above for a particular time. This

gives you the capability to have pretty close to 100% coverage of your network performance and be able to zoom in on any particular outage, period of slow response, etc.

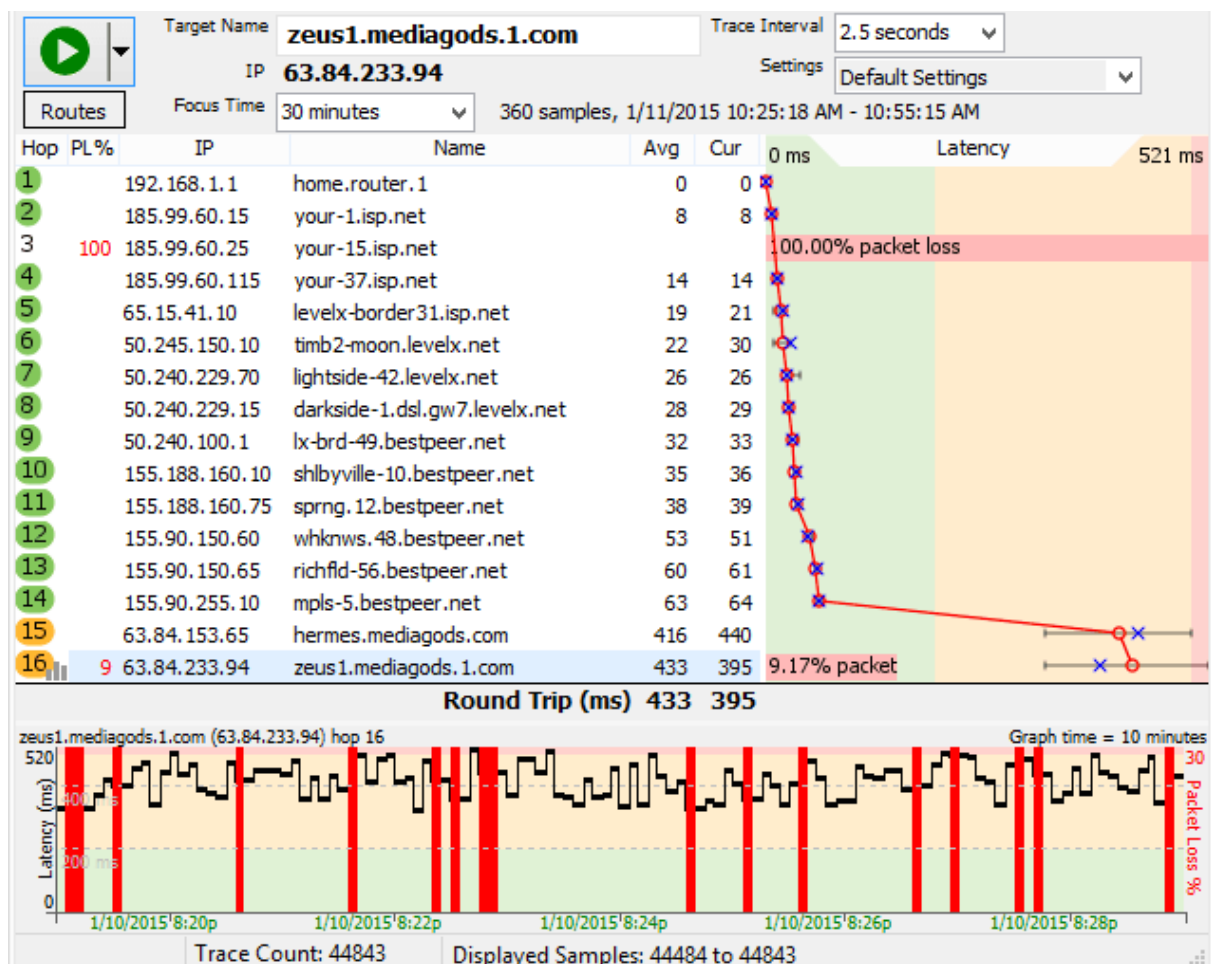
The options in the alert setup do allow you to have the .PP2 file (trace data) emailed, and then you can use these same capabilities to zoom in on that data.

## 6.6 Interpreting Results - Gamers

For this example, let's assume you're an online gamer - specifically, a Quake III player (though the following is representative of any online game really - MMORPG, RTS, racing sims, fighting games, etc.). You've got two servers that are running the same maps you like to play, so the only issue you have is which one out of the two is going to give you a better connection. We realize some folks aren't going to be so patient as to use the method below to decide which server they're going to play on... but bear with us here. We're learning! The same topics we go over in this section, as far as graph interpretation, are also applicable if you were trying to figure out why a connection to a specific server you were just playing on is so cruddy.

The first thing you need is the IP addresses or DNS names for the two servers. You'll launch PingPlotter, enter the IP address of the first server into the [Address to Trace](#) box and click the Trace button. 2.5 seconds is a good value for the [trace interval](#), and the [# of times to trace](#) should be unlimited (we're gonna watch it for awhile). Then, start a trace to your second server's IP address using the same settings you used for the first server.

You then go get yourself a Diet Pepsi, do some stretches or whatever in preparation for a night of gaming. When you get back to your computer you'll have two graphs that look similar to the ones below. Let's analyze both and see which server you want to play on.



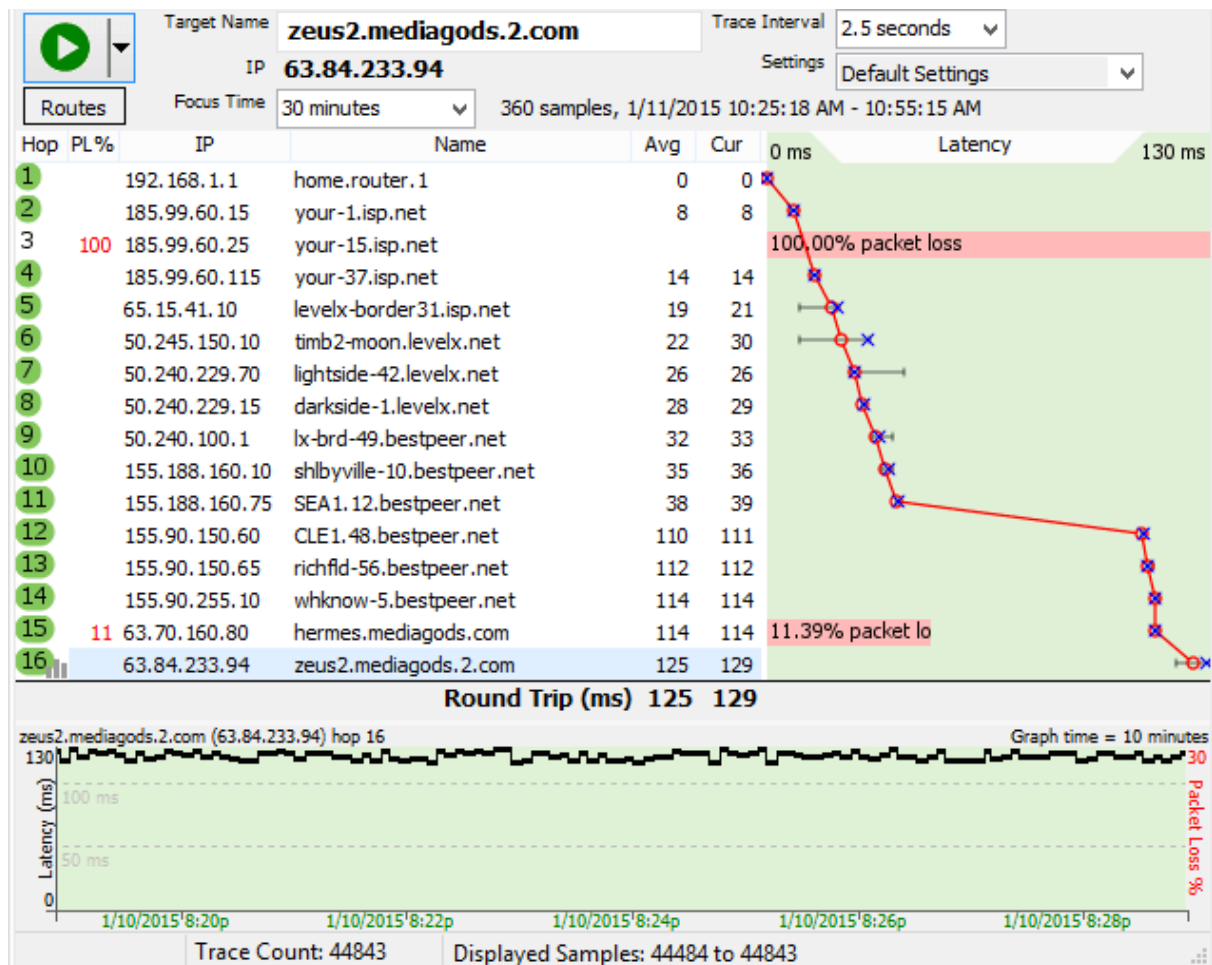
Hmm. This doesn't look too bad until you get to hop 15 and start looking at the history graph. Let's take the [time line graph](#) first.

**Red is bad.** Every time you see red on the history graph you had a [timeout](#), or in other words there's dropped packets. Packet loss is the bane for most online games. When you're running across a big open area and then all the sudden \*blip,\* you're on the other side (and most likely dead), that was more than likely caused by timeouts, or packets you didn't get to (or back from) the server. Besides the red lines on the history graph, you can also see your packet loss in the PL% column and, if you look at hop 16, the horizontal red line contains your packet loss value.

Digging a bit deeper, you can see that you're running under 100ms all the way down until you hit hop 15. Notice that you move off of LevelX's backbone into BestPeer between hops 8 and 9. No problem there, there's plenty of bandwidth between those two providers since the time doesn't really go up. From the DNS on hop 8, we can see that hop is a gateway (thus the "dsl-gw7" part of the name) to some DSL customers. Where we start running into problems is when we get off of BestPeer and hit the Mediagods domain that's hanging off the DSL link. All the sudden your latency goes up to 400+ms at hop 15. That

DSL connection is busy. Once you make it to the server at hop 16, not only is your latency still way up there, but you're getting 9 to 10% packet loss. That server is a busy bee also it seems. So busy that he's not keeping up. Combined with the bandwidth saturation we're getting on the DSL line itself, it's best to try later. We don't want to play here.

Now let's look at our second server.



Now this is more like it. Really, anything under 150 ms is a great connection. We don't even make it over 40ms until hop 12. Sweet.

Let's look at that connection between hops 11 and 12. Notice from the DNS names that you actually go from Seattle across Global Crossing's backbone to Cleveland. When you factor in speed of light latency, you can account for about 40-50ms of your latency to hop 17 with that hop across the backbone between hops 11 and 12. So you've got a 120-130ms ping to hop 16. That's pretty good. If you didn't have that fat pipe installed (and were instead running a modem) you'd probably be running at about 220-230ms for your latency.

"What about that 11% packet loss at hop 15?", you ask. Judging from the numbers for that hop and hop 16, what we're most likely dealing with here is a router that probably has a low priority for ICMP packets. A lot of network admins will set a router up to drop ping/ICMP packets first if it starts getting busy - [have a look here for more details](#)

So which server are you going to play? Obviously it's the second server above.

## Other considerations

Your graph results can be affected by a number of things that are out of PingPlotter's control.

1. In the analysis of the second graph above, we mentioned that hop 15 is more than likely just a problem with that router not giving us back good information. Many routers put a low priority on [ICMP traffic](#) . Others don't even echo back ICMP requests (this will show up as a blank entry for that hop). Obviously, PingPlotter has no control over these situations.
2. PingPlotter can't track the route that your traffic takes on the return trip from the server back to you. If your inbound traceroute traffic is taking a different route back to you than the outbound traffic to the server, this is called an **asymmetrical route**. By definition traceroute doesn't take these types of routes into account and, unfortunately, PingPlotter isn't going to be able to tell you about problems with the return route in these cases. One clue that this is happening is that you'll have a great trace up until the last one or two hops on your trace. In other words, you don't have an easily identifiable problem at hop X further up the route that is mucking up the rest of the route downstream.

One thing you can do is save your trace data to a text file and post it up on a support message board for the particular game that you're playing. Even better, save off a graph and post it instead. Many sawy game server admins will actually do a traceroute (or even better a PingPlotter trace... \*smile\*) back to you and be able to tell you if there's problems with a route back to you when asymmetrical routes are involved.

There are some sites that can do traceroutes back to you if you want to investigate on your own. They can be found [here](#).

In closing, we can't emphasize this enough: **latency is the bane of online gaming**. Much more so than bandwidth limitations. The good thing is that PingPlotter can tell you this latency, and provide you with ammo in the way of trace and graph data when you're beating up on your ISP.

## 6.7 Interpreting Results - ISP Problems

For this example, we're assuming the role of a user that's having problems with a broadband connection.



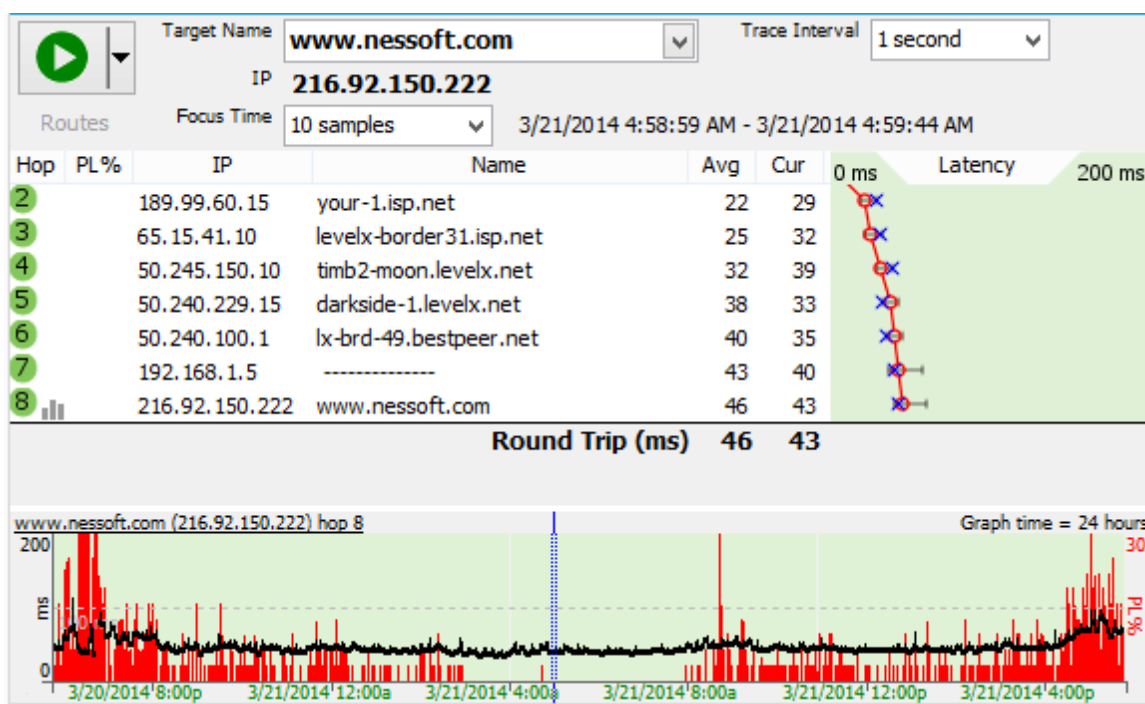
What we'll be taking a look at is a few days worth of trace data.

Before continuing, if you're not familiar with how the graphs work in PingPlotter please make sure you've read the [introduction to graphs](#) earlier in this tutorial.

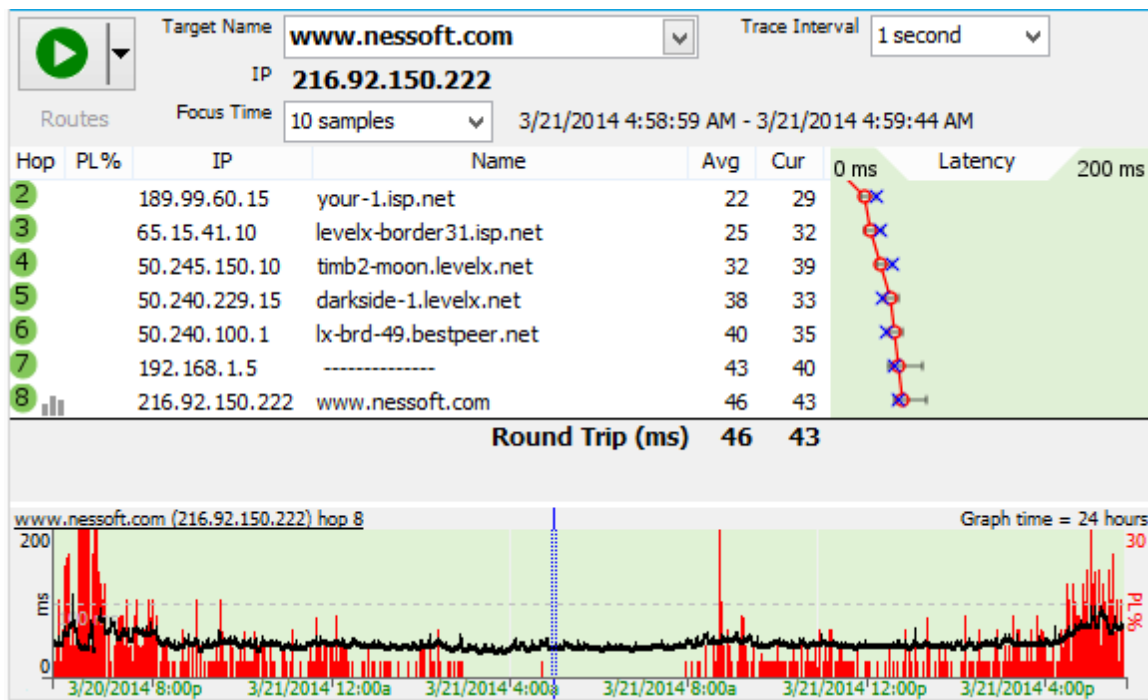
One common mistake we see folks make is that they'll trace to their ISP's border router. This is a bad thing. If you're tracing to the border router and your route changes (i.e. they take that router down for maintenance or you get load balanced onto another router) you really have no idea what happened. If you want to keep your traces local to your ISP, trace to an address that isn't going to change on you like you're ISP's mail server. This is actually a good thing to do if you're having mail problems and it's your ISP's mail server going down. Otherwise just pick a destination that you know has a reasonably good chance of always being up. This is a better choice since routes within your ISP can change, and PingPlotter keeps track of those route changes. The cool thing is that you're doing a traceroute here, not a ping, so even if that destination host goes down you can drill down on the timeline graph and see if it's your connection, or if it's just the destination being down (as in all hops but the destination don't show timeouts).

Another problem with tracing to your ISP's border router is that your ISP will not *respect* the data that you collect this way. No application targets a border, so they have no reason to trust this data. For best results, you want to pick a target that is one you use and are having problems with.

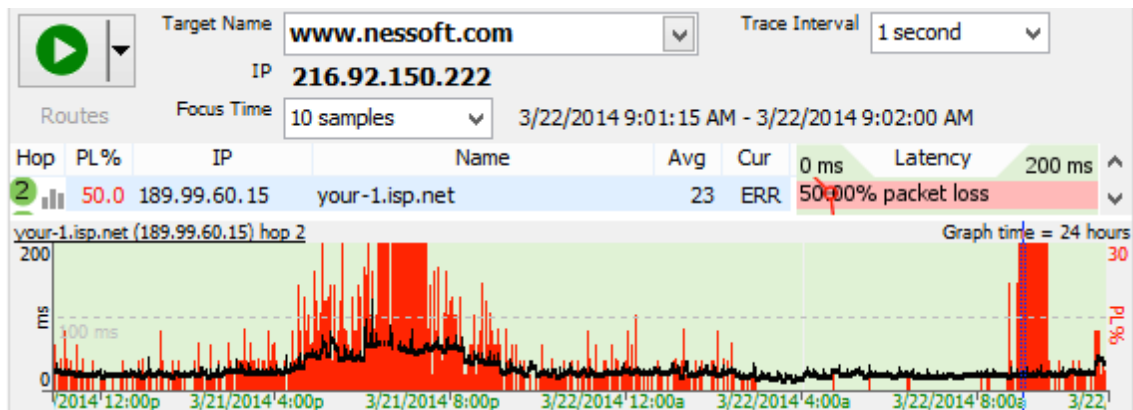
Note: For clarity, all the graphs below show us ignoring Hop 1 which you to can do from the View Menu. All the graphs were saved with the File/Save Image command within PingPlotter then converted to .gif for this tutorial.



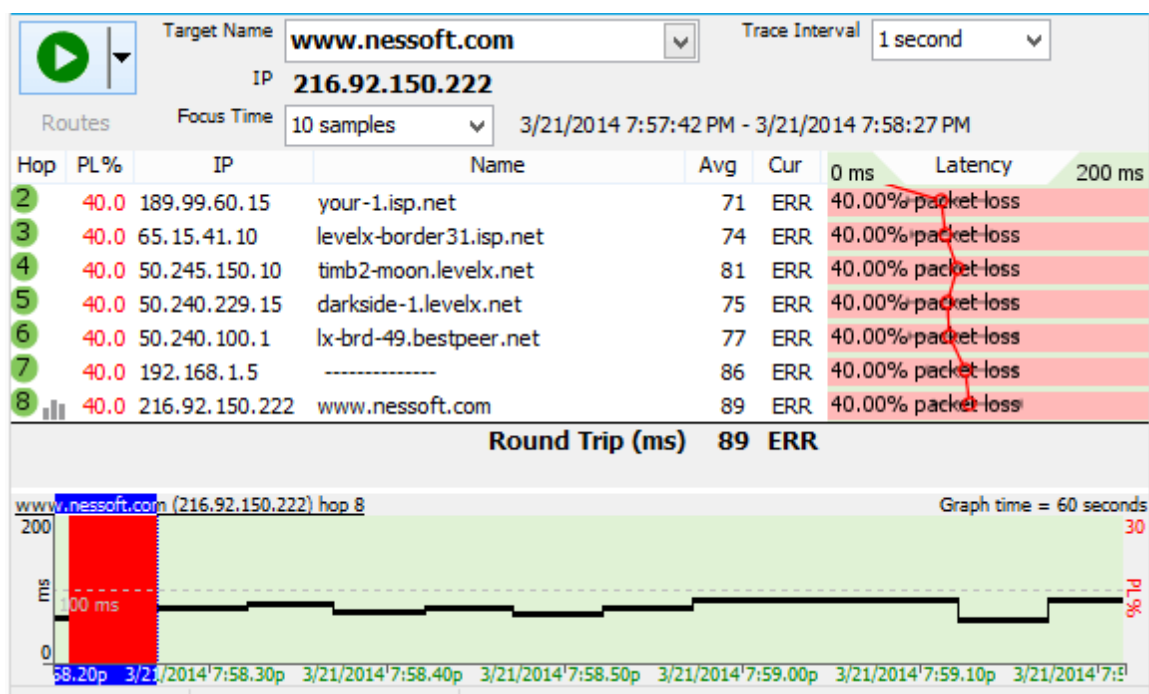
This first graph shows what the traceroute should look like with no load on the connection, i.e. no downloads, streaming audio, on-line game playing, etc. "What about all that red on the history graph? I thought red was bad?", you ask. Actually **red is bad**, however before we saved out this graph we **double-clicked on the timeline graph to drill down, or zoom-in**, and are looking at the data for 3/21/14 at 4:58 a.m. If you look at the top of the graph you see the "Focus Time" is set to "10 Samples," and the range is between 3/21/14 4:58:59 AM - 3/21/14 4:59:44 AM. So basically the above graph's trace for that *particular time* looks good. However when you look at the timeline graph, you can start to see the tale of woe. What we have here is a really flaky broadband connection. So how do we prove it? Read on.



Just sending your ISP a graph with red lines isn't very convincing. However, when you start zooming in on those sections with timeouts, and send graphs of them as well like this second saved graph, it's pretty obvious the connection's hosed when you can't see out to Hop 2. This is the same time interval as the first graph, just showing a different period in time for trace data.



For our third graph we've got data for early on the second day of our trace. Lots of red, and when we focus on the 9:01 AM time period the connection's still poor. We're unable to see out most of the time. It's hard to argue with the graph. Also keep in mind that what we're showing here is that the whole timeline graph isn't solid red. This isn't an issue where you accidentally kicked the plug on your router.



For our fourth graph, you can see from the timeline graph's times that we've adjusted the time-interval so we're only looking at 60 seconds worth of data. The trace graph is showing the section of the time-interval graph that we double-clicked on which is 7:57:42 PM to 7:58:27 PM. So what's up with the 40% packet loss showing up on the trace graph? Notice that we didn't have timeouts for about a full 50 seconds (out of 60) on the timeline graph. Out of the ten samples we're looking at, 40% of them were timeouts. This is important. When we're looking at the trace data we're looking at those 10 samples we selected and the numbers for those samples, not the whole range of data shown on the timeline graph! This is not a graph you want to send to technical support. All it's going to do is confuse them.

So in summary, PingPlotter allows you to show your ISP where the problem's are. In the these examples, we were essentially showing the whole link going down. However, we could've just as easily seen if the ISP's connection to the Internet was down at Hop 4, because we were tracing to a destination not on our ISP's local network. If there was indeed a problem at Hop 4, we would've had good trace data at Hops 2 and 3, timeouts at Hop 4 and possibly no trace data past Hop 4. If the router at Hop 3 was being flaky, and for instance you saw a lot of packet loss, it's easy to save an image showing just that so you can email it. When sending graphs to your ISP, we've found it's best to send one graph showing data for an extended time period, and then drilling down on the timeouts and sending graphs that truly show them what's going on. PingPlotter allows you to save in .png or .bmp format. We recommend .png because they're smaller.

## 6.8 Interpreting Results - Bad Hardware

### **Scenario: External customer has problems using your network resources.**

A customer (not inside your network) has problems losing connecting to services inside your network. In this scenario, you are acting as a service provider for some network service. This might be provided via HTTP, or possibly through something like Citrix or Windows Terminal services.

Your customer (possibly employees of your company, or maybe subscribers of your service if you're an ASP-based service) is complaining of frequent disconnects, and possibly slow performance sometimes. How do you troubleshoot this kind of a problem? Where is the problem; at the entry point to your network, in the customer network, or possibly in one of the providers in between?

One way to pinpoint the problem is to have your customer run PingPlotter against your service. They can easily download PingPlotter, capture data, and then either analyze that data themselves, or email the data back for you to analyze. This information can be used to pinpoint which hop (or router) in the chain is adding latency or losing packets that might be causing problems back to your ASP.

### **Analysis:**

Here's an example extracted from a real-world customer situation. We'll walk through some of the symptoms, collected data, then analysis and how we came to the proper conclusion.

Our ASP (we'll call the ASP "CitServeCo" - a totally fictitious company) is accessed via Citrix, which is relatively sensitive to high latency and packet loss. The customer (who uses financial applications served by CitServeCo) was frustrated by disconnects during the day, which lasted anywhere from a few seconds to a minute or two, interrupting their ability to do business.

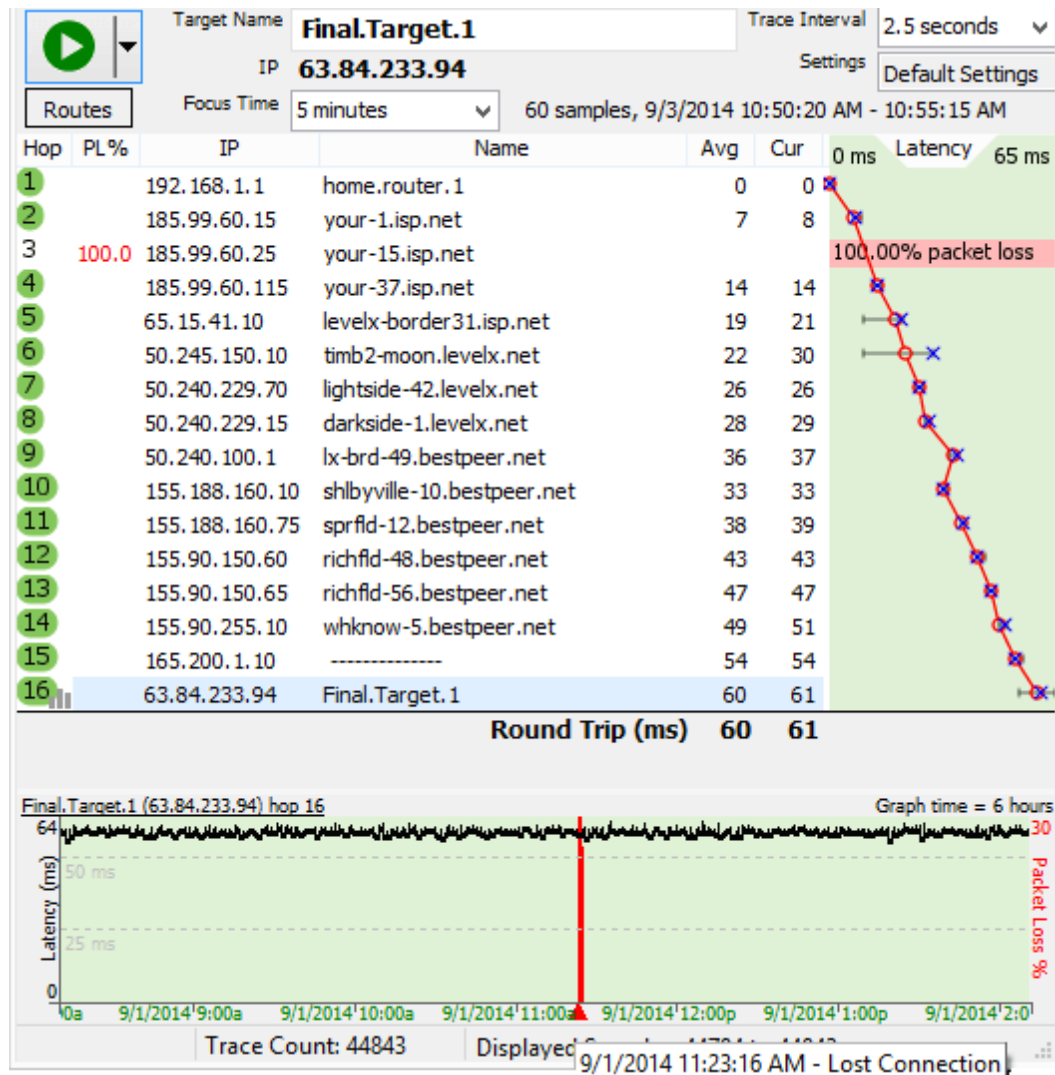
The customer will almost certainly blame CitServeCo initially for a problem like this – but we need a way to determine where the problem is being caused, and help the customer solve the problem. Any reliable ASP will get numerous complaints like this – that they \*know\* are the fault of something beyond their control (like the customer's cable connection, or similar). Most ASPs will attest to the fact that customer connections are a prime source of network problems, but we can't just tell the customer "It's your problem."

When the customer contacted CitServeCo, they suggested that they download PingPlotter, install it onto the workstation they normally work with, and configure it to monitor the Citrix server inside CitServeCo's network.

If a disconnect or slowdown occurs in Citrix, we suggest to the customer that the right-click on the lower “time-graph” in PingPlotter and [create a comment](#) at the point that the problem occurred. This correlates real world symptoms with the collected PingPlotter data (a crucial part of the troubleshooting process).

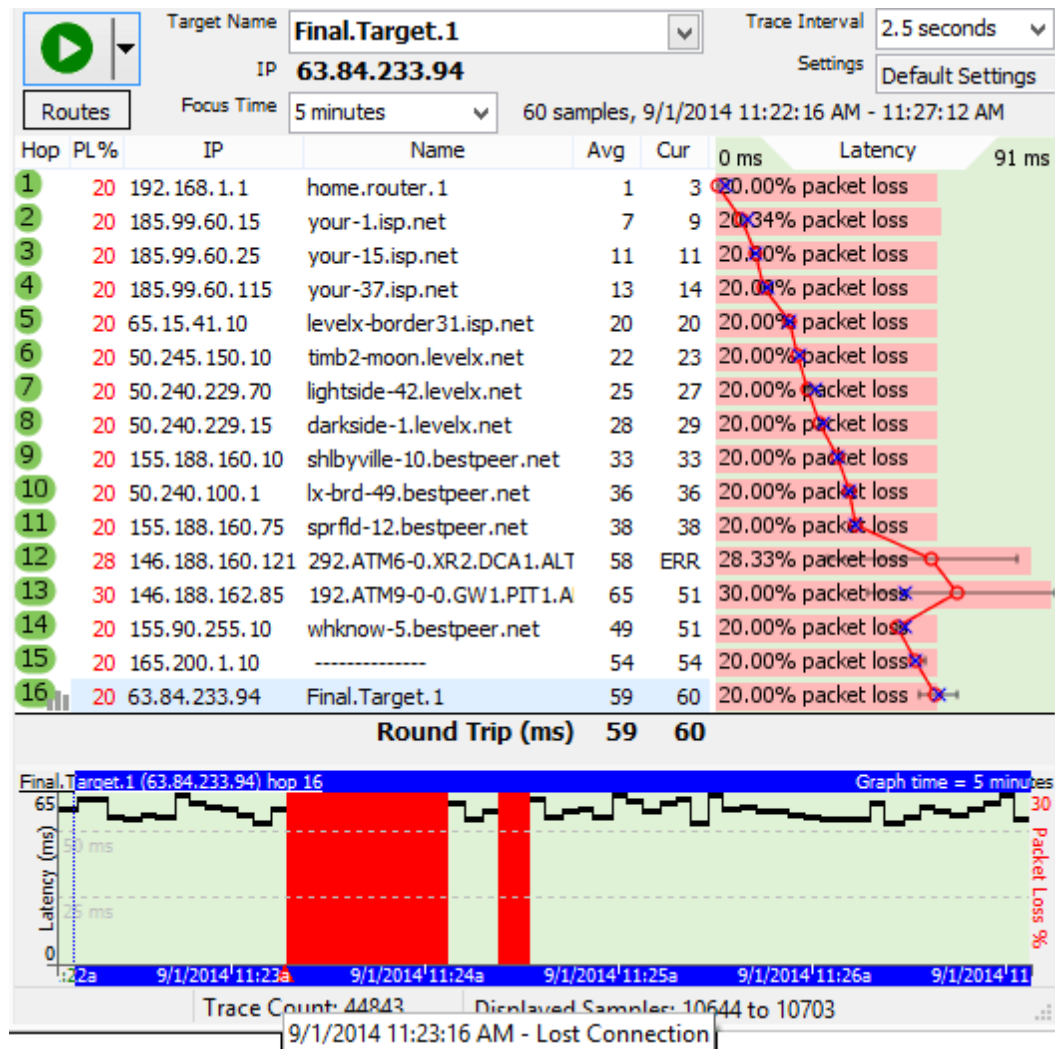
A day of collecting data resulted in only a single disconnect, and the customer dutifully recorded that for us. They then sent us the saved data for analysis. Because PingPlotter data is already compressed, there is no need to .zip up the file, making it easier to send data.

Here is the result of that day.



Notice the packet loss in the timeline graph at 11:23 am. This corresponds with a note from the customer in the data file that shows the disconnect happens. We decide we want to take a closer look. To do this, we set the “Focus Time” to 5 minutes, double-clicked on the red point in the time-graph, and

then reset the time-graph period to 5 minutes instead of 6 hours.



Wow. A big outage. If no packets were getting through, we're definitely going to see problems. Where did that loss occur? If we look at the upper graph, we see that we have consistent packet loss across \*all\* the routers (double-clicking on the lower graph focused the upper graph on the period we were interested in). The packet loss was similar, but let's have a look at the actual lost packets. To do this, double-click on a router in the upper graph to show a time graph for that router.

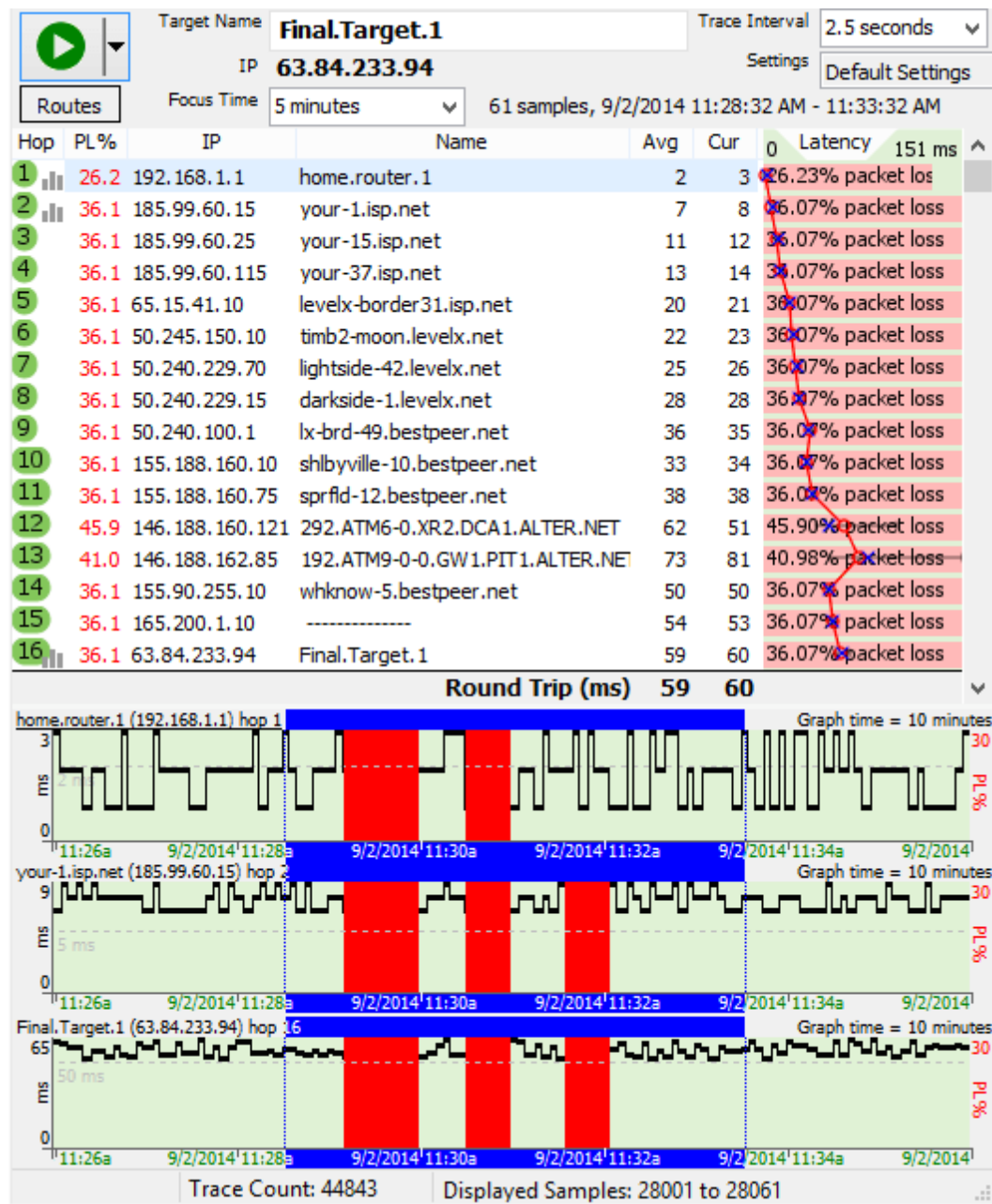
It looks like hop 1 lost just as many packets as hop 16. Since every piece of data needs to go through hop 1 to get to any of the other hops, a blockage there will look just like this. It looks like we've found a likely culprit – the router at hop 1, or possibly anything between the computer collecting data and hop 1. This might be as simple as a network cable, or it might be a significant amount of network equipment. We don't know until we check with the customer to find out what's here.

We asked the customer what kind of network hardware they have in place. While they were collecting this information for us, we had them continue to monitor their connection.

It turns out that all the computers at this location are hooked up to a "SOHO" router. This SOHO router is, in turn, connected to a cable modem provided by their ISP. From their ISP, the customer is uncertain as to the network configuration. We see some of this in the PingPlotter graph – a list of routers that are participating in sending data.

The customer was uncertain as to what the individual pieces of hardware reported in as, so we had them continue to monitor, but asked that they cycle the power for the devices if they had a disconnection problem. Several days later, they had an opportunity to do this, and captured the experience in PingPlotter.





In this instance, the customer had a disconnect, and then rebooted their SOHO and their cable modem. Notice how hop 1 stays working through the reboot of their cable modem, but not through the reboot of the SOHO. So this must be the SOHO. Also, notice during the disconnect, hop 1 was non-responsive. This indicates the problem was similar to having the SOHO powered off.

Maybe there was a bad power supply on the SOHO, or maybe a bad network cable someplace? The customer replaced the network cables, and the problem persisted. They then replaced the SOHO router

and the problem was solved.

It's not every case where the problem is so "in our control" as it was in this one. Sometimes, the problem requires talking an ISP into replacing some piece of their hardware, or replacing an ISP altogether.

PingPlotter can be just as effective locating the outages, packet loss, or latency on ISP equipment, though – it's can sometimes just be harder to fix those problems.

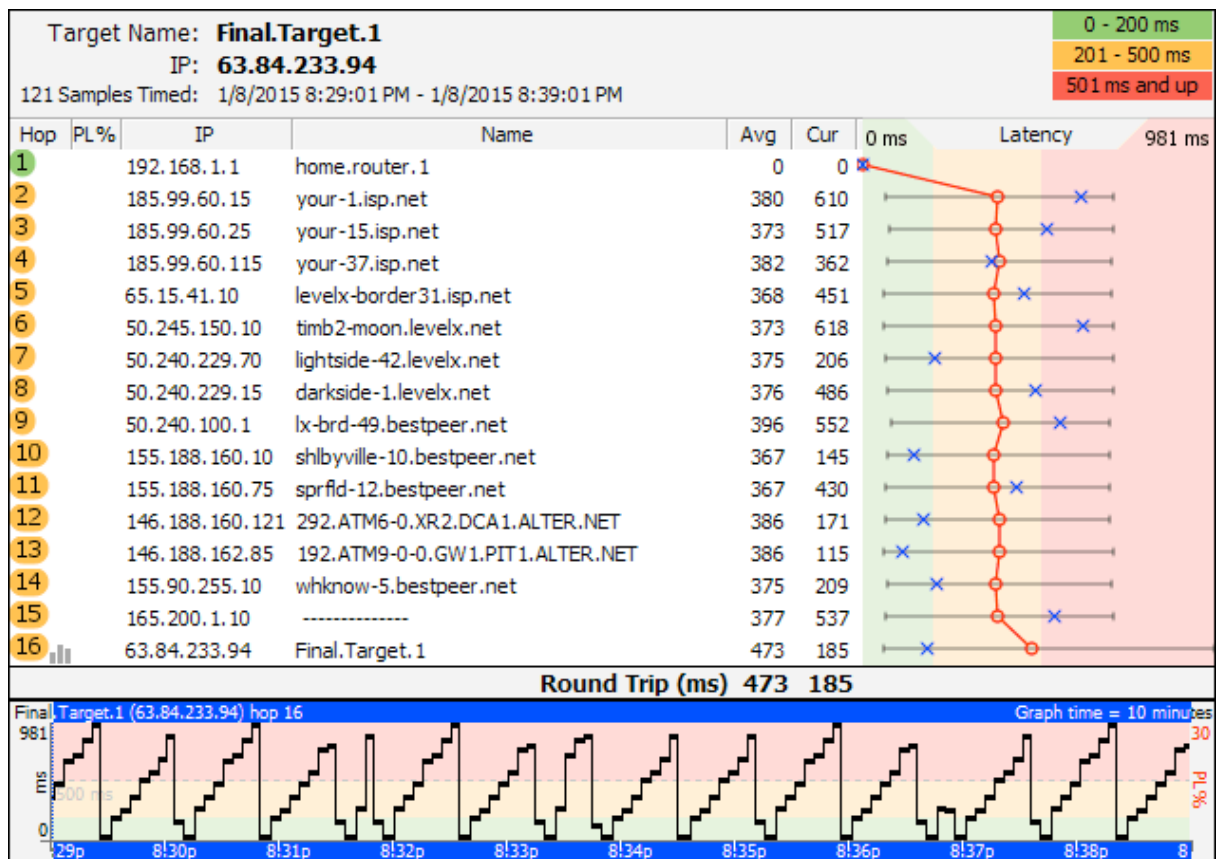
## 6.9 Interpreting Results - Bandwidth Saturation

Every network connection has a limit, and in this example we'll talk about how to recognize bandwidth limits on a local DSL / Cable modem.

The scenario here is a 30Mbps downstream, 10Mbps upstream cable modem running a 2.5 second trace interval to our target. The computer running PingPlotter was connected to the cable modem via a wireless network card, which loses packets occasionally.

We're downloading a 2 gig file, and during this period the bandwidth of the cable modem should be completely saturated. This means that anyone else using this network connection is going to notice significant latency and possible packet loss. Any user might decide they want to troubleshoot this situation, and might run PingPlotter to do so.

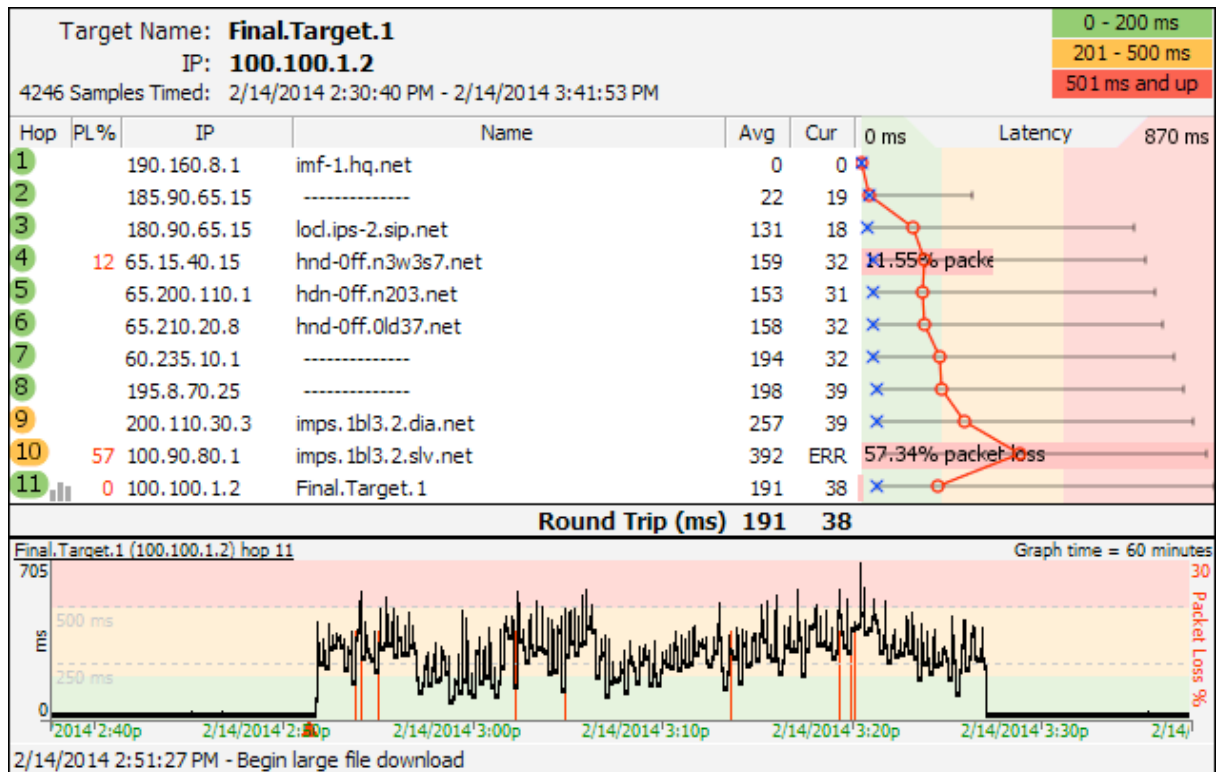
A quick 10 minute trace might look something like this:



First, notice the big latency jump between hop 1 and hop 2 - the DSL modem. This same latency jump is translated into all downstream hops, which indicates that hop 2 (or the link between hop 1 and hop 2) is significantly impacting network latency. Since we know what's going on here, it's pretty easy to recognize. In a lot of cases though, you might not know what's happening and you'll see latency like this.

Now, the "Sawtooth" pattern in the timeline graph is a classic bandwidth saturation pattern. Anytime you see a pattern like this, think "bandwidth saturation". Keep in mind that bandwidth saturation is totally normal, and happens on just about every network. Consumer-grade DSL and cable modems have the highest chance of seeing a pattern like this, but it can happen on any connection where the bandwidth at a network point is being totally saturated.

Here is another example of bandwidth saturation (with more time shown on the graph):



Here you see an 40 minute period where a large download was happening. Outside the download, the latency is pretty much rock solid. The packet loss stays relatively constant across the entire period, indicating that the packet loss isn't being caused by bandwidth saturation. We know that it's actually caused by a wireless network that has about 1% normal packet loss, and the download rate doesn't affect it at all.

There are several solutions to the latency problem:

- Procure (buy) more bandwidth
- Don't transfer as much data
- Implement QoS

This same situation happens in any network. Your ISP runs into these same exact options with the connection between their network and the outside Internet. They can limit what you download, or they can buy more bandwidth. On a bigger pipe the jump in latency isn't going to be as pronounced, and the bandwidth will be saturated by a number of different loads. Often on a primarily consumer-use network (DSL / Cable), the increase in latency will be gradual as more people use the bandwidth, and then it will gradually drop off as people stop using it. The rise in latency often happening at 6-9pm, and the drop off happening at 11pm - 1am. It's often accompanied by packet loss as well.

On your own network, the bandwidth saturation could be happening for a number of reasons: another user downloading something; temporarily restricted pipe size (problem with your Internet connection), p2p applications in use, or possibly even a virus/worm type application that is using bandwidth sending out new instances of itself.

In summary: Saturated bandwidth can be normal, and isn't always an easy problem to solve. It's often easy to recognize the symptoms though, especially if you look at the trend over time. PingPlotter can really help pinpoint the problem, giving you the ability to see trending over time, latency patterns short and long-term, and latency / packet loss correlations.

## 6.10 Voice over IP (VoIP) troubleshooting

### Introduction

Using an IP Network (like the internet) to conduct a voice conversation ([VoIP](#)) is becoming easier and easier for people to do. It can be inexpensive and relatively reliable. It can, however, also be challenging - with poor voice quality, the inability to hear and communicate, delays and other problems.

The underlying technology for VoIP is extremely network dependent. If you're having voice quality problems, the problem is often related to the network - maybe your internet provider or maybe some other component between you and the called party. This article will talk about some basic troubleshooting techniques that can be used to locate where the problem is occurring so you can make good decisions about how to solve the problem.

### Network-related VoIP Symptoms

Many symptoms of VoIP problems are network related (although certainly not all of them). Here are some examples of symptoms that are often network related:

- Garbled words ("blips" and clicks mixed in with the words)
- Parts of words missing
- Gaps where the other side is talking, but you don't hear any of it
- High "distortion"
- Delays between the time you talk and the other side hears you (and vice versa)
- You start talking not realizing that the other party has started talking already too, and you talk over each other for a few seconds

Other symptoms might not be network related. In particular, if the symptom *\*always\** happens, any time

of the day, any day of the week, then there's a decent chance it's not a network problem.

- Echo when you talk. This can be exacerbated by network problems, but constant echo is usually not caused by network problems.
- Inability to connect a call to some users while you can call others

## Using PingPlotter to identify the source of network problems

PingPlotter has some unique capabilities in its ability to help you track down the source of network problems. What you really want to know is:

- Can **you** fix the problem or do you need to call someone else to fix it?
- If you need to call someone else, who do you call? Your ISP? Your VoIP provider?
- When you call them, how do you convince them it's their problem to fix?

PingPlotter can offer a **lot** of insight into all of these questions.

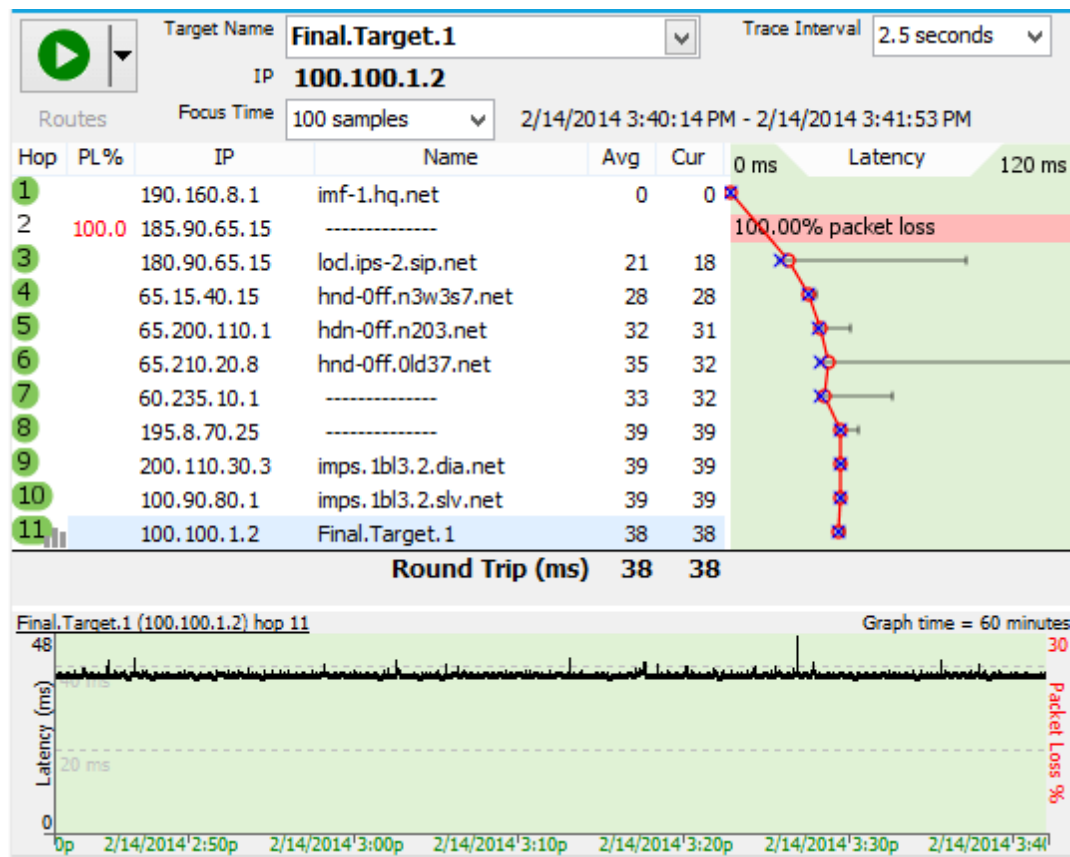
## Collecting data with PingPlotter

Before we can do much analysis, we need some data to analyze. We cover some of these topics in earlier manual entries, so we won't cover \*details\* of how to set things up here.

First, we need a target server to monitor. Ideally, this would be the actual VoIP server of your VoIP provider, or something on the same area of the network. If you called your VoIP provider and they asked you to collect PingPlotter data, they may have given you a server to use. In many cases, the use of **any** server can work, but this will only identify problems with your ISP - not with your VoIP provider. The good news here, though, is that the vast majority of VoIP problems are because of front-line service providers (like your ISP). If you don't know what address to use and you have no way of finding out what address your VoIP hardware or software is using, try using the web site of your provider.

For this discussion, we will be a fictitious server (Final.Target.1) as the server we're monitoring and using for troubleshooting.

In your instance of PingPlotter, enter your VoIP server (or related target in the "Target name" field, and set the trace interval to 2.5 seconds. Now, hit the "Trace" button (or the "enter" key on your keyboard), and you should see a picture appear that looks something like this:



The upper graph should show a full route, including the "Round Trip". If you don't get a Round Trip, check in the troubleshooting section of this document for some ideas.

Now, let this run for at *least* 30 minutes - preferably, during a period where you're making a voice call. Ideally, you'll have a period where you have a voice call that's good and one that's bad, but that might be possible. If nothing else, just let it run for long enough to get a good sample of your network conditions.

A great thing to do while you're collecting data is to make notes in the PingPlotter data about what you're experiencing. You can see instructions on how to do this in our [Timeline Graphing](#) entry under the "Creating Comments" section.

The data we collected covers several days. PingPlotter works great to just run over a long period of time so you get a good idea of what network conditions look like - during good times and bad.

## Examining data with PingPlotter

Once you've collected some data, it's time to have a look at what might be the problem. We cover some of the PingPlotter commands on zooming, focusing and digging in the [Finding the source of the problem](#) section.

One of the key things to know here is that we're looking for problems at the last hop only - and then using the other hops to determine where the problem starts. Packet loss or latency that [shows up only at an intermediate hop is not a problem!](#)

Let's look at the graph above. Notice how hop 2 has a full 100% packet loss? The final destination looks rock-solid, though - no packet loss and the latency, and it is mostly nice and smooth. This is, in general, what you want to see - a mostly flat line at the final destination, no packet loss (red lines in the time graph).

## **So, what are we looking for, when it comes to problems?**

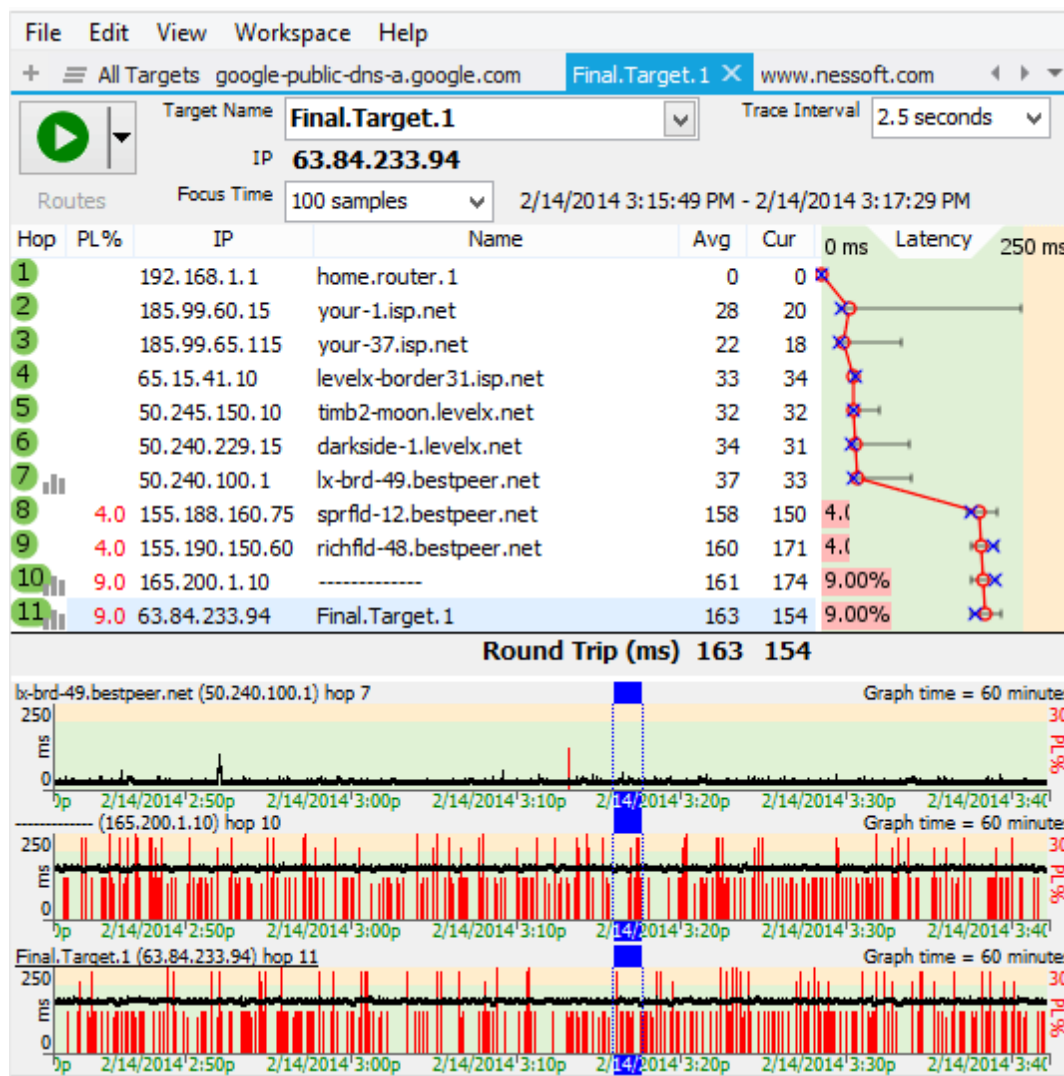
The first place to look is the final destination. If the time graph looks like the graph above (straight line, no red), then PingPlotter is not finding network problems. Look for problems at the final destination. If you find a problem at the final destination, then look back until you find the first hop showing similar symptoms - that's who we probably need to contact to get the problem corrected.

## **Examples and Analysis**

### **Example: Distributed packet loss**

Let's look at an example, this time an example with problems:





Here, we see 9% packet loss at hop 11 (the final destination). This would result in poor voice quality, dropped "bits" from words and hard to understand conversation. Notice that the latency is pretty good still - it's just the packet loss that's a problem (packet loss is all of the red in the time graphs and the red bars in the trace graph). With a pattern like this, voice quality would be consistently "iffy" - not unusable all the time, but not very good either.

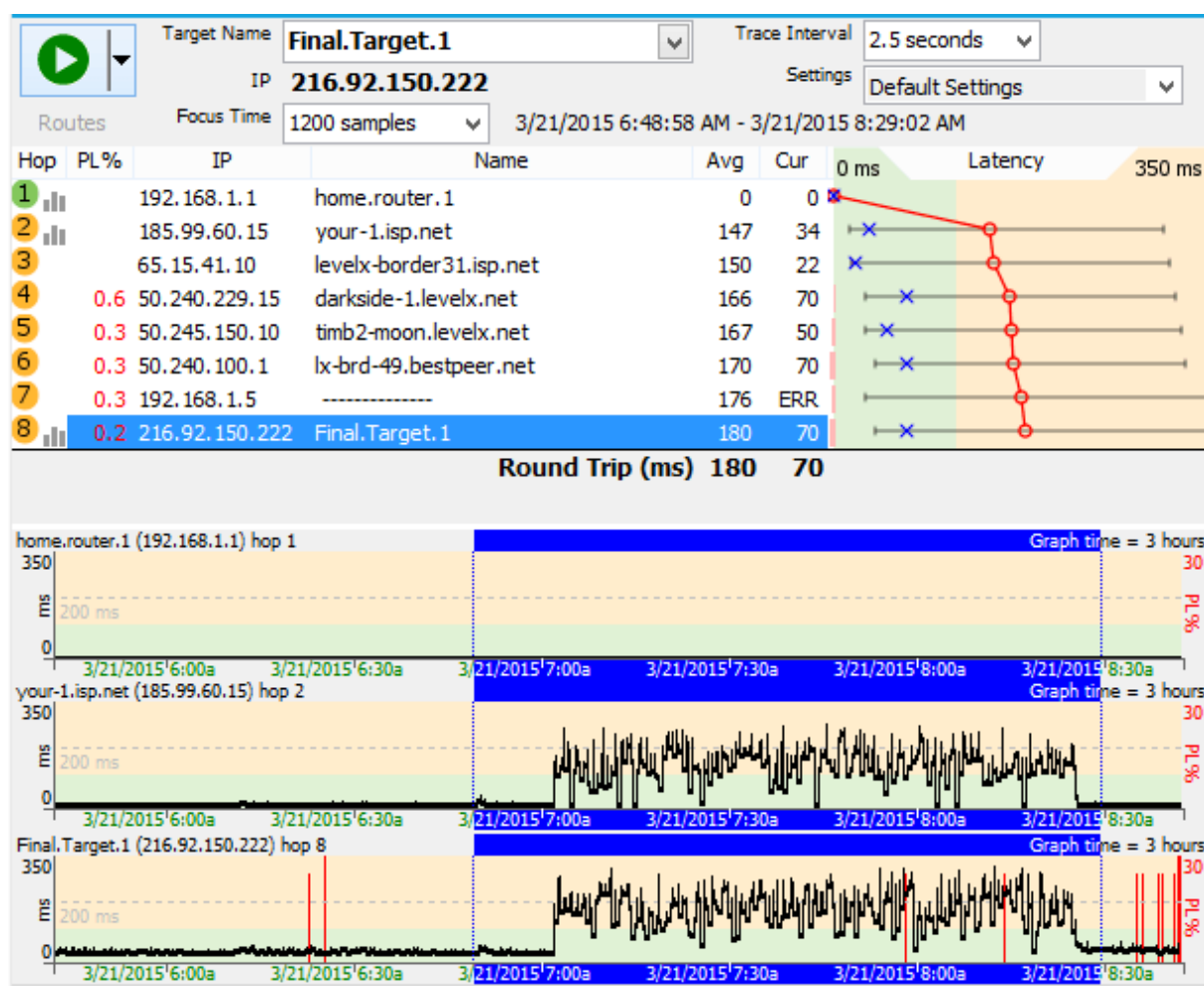
Notice how the packet loss is happening at all hops from hop 8 onward, while hop 7 looks relatively good. The packet loss percentage fairly similar all the way down (although, statistically, it would be just about impossible for all hops to have identical packet loss percentages with this kind of loss). To turn on and off time graphs like this, just double-click on the hop number in the trace graph.

So, in this situation, the problem looks to be between hop 7 and 8. It's pretty likely that Bestpeer knows about this problem - it's in the "middle" of their network - and it's all owned by Bestpeer (we can see that

from the DNS Name column).

In this case, we would need to contact Bestpeer about this problem. The picture above is pretty compelling and would be a good communication tool to them.

## Example: Local bandwidth saturation



Here, notice the big latency jumps - you have a nice flat line, then a jump in latency, including some packet loss. This pattern is one that is almost always a bandwidth saturation issue (which is the same as congestion). In the case we have here, hop 1 is inside our network (our DSL modem, actually) and hop 2 is inside of our ISP.

This is a case where we were transferring too much during this period - and we were using all of our available bandwidth. A VoIP call would suffer significantly during these periods - there is a lot of jitter (the "ragged" line is an easy way to see jitter - where packets take different amounts of time to arrive), higher latency and some packet loss. The voice quality would be bad, there would be additional lag, and it

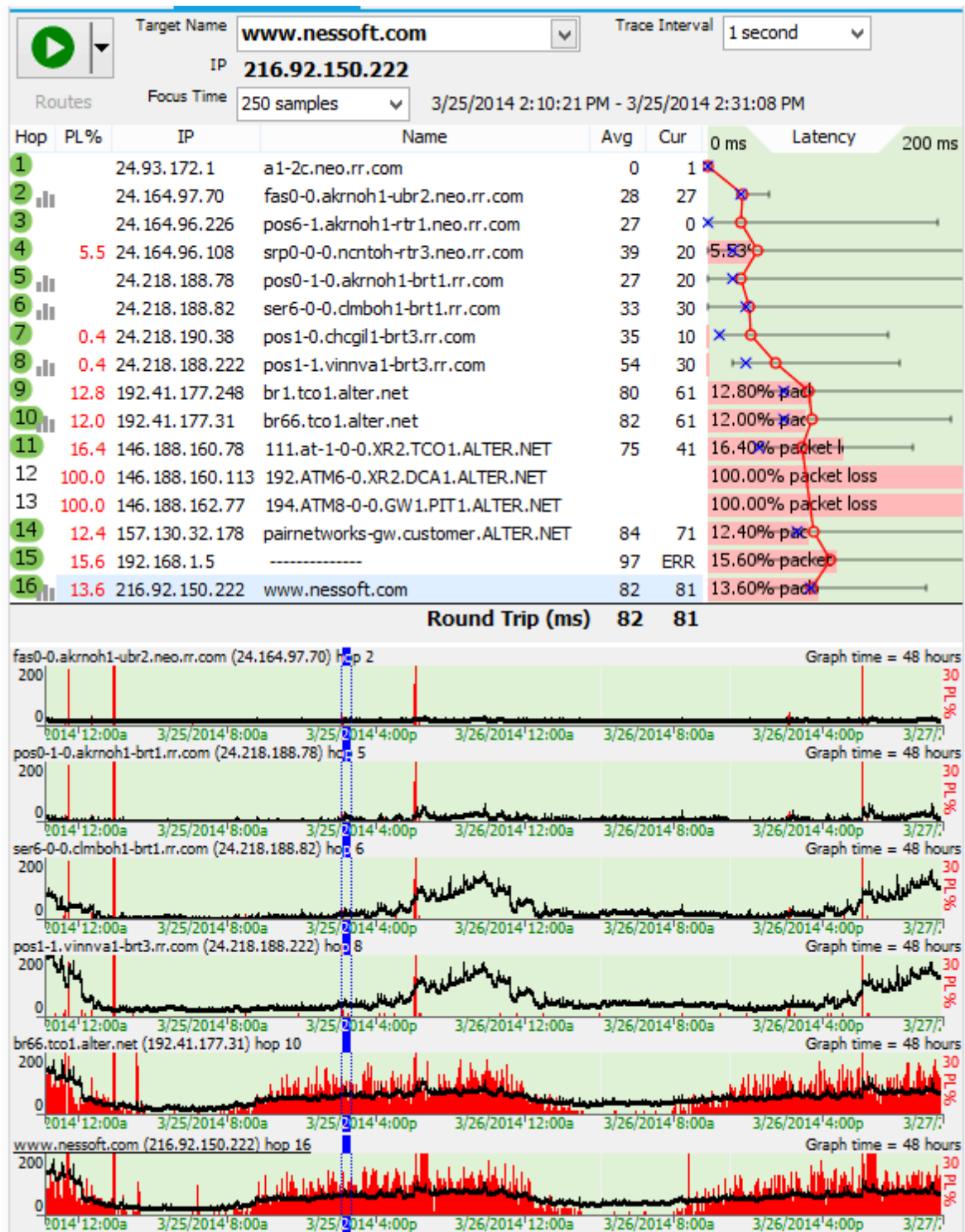
would probably have audio drops.

There are a few options for solving this one, but none of them involve complaining to anyone else:

- You can install a traffic shaping modem that gives higher priority to VoIP data (actually, unless you've configured PingPlotter packets to look like voice data, you might already have one of these in place - this article does not cover that topic, though).
- You can get more bandwidth (although that doesn't solve all problems - as you'll still be able to use all your bandwidth).
- You can use less bandwidth.
- You can get an additional broadband connection and dedicate it just to VoIP (this is an especially good idea for heavy VoIP users or businesses). The low cost of an additional broadband connection makes this viable in a lot of situations.

## **Example: Border congestion**

Congestion often happens at network borders - where one network hands off to another. This is relatively common for small, growing ISPs - where they just do not subscribe to enough bandwidth to handle everything. Let's have a look at what this condition might look like.



This one isn't \*quite\* as simple, as there are a few factors. The symptoms of conditions like this would be:

- During high load times, this connection would be completely unusable for VoIP. With 14% packet loss

and really high jitter, this would be absolutely horrible for any kind of voice call.

- During the early mornings and late evening, it would be \*bearable\*, but the jitter would cause the words to be garbled sometimes, and not too fun.

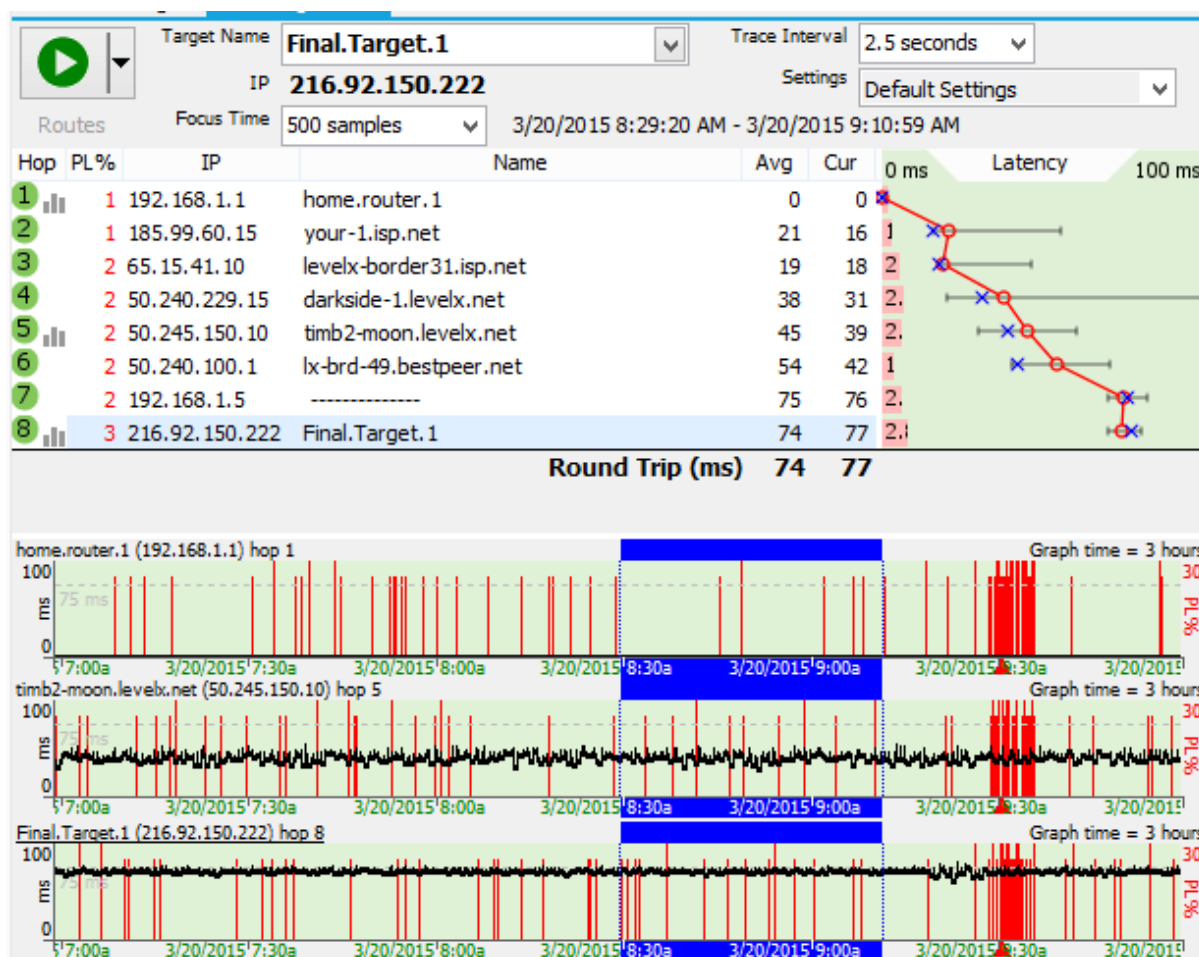
If we look at the network conditions with PingPlotter, we see a couple of problems. First off, there's some serious packet loss starting at hop 4. This packet loss is carried down through the rest of the route to the final destination. This is a border - between our ISP and levelx.net. Having problems at borders like this is pretty common - that's where one company pays another to handle traffic. If a company is growing, it might be "oversubscribed" - using more bandwidth than is available.

An interesting part of this is how during heavy load times for home users (ie: evening hours), the packet loss and latency are worse. During early morning hours, it goes back to being OK again. This is a big sign that the problem is load related - and that this link is having congestion problems at "rush hours". Time to add some lanes!

Another problem, though, is inside the rr.com network there is significant latency and jitter. There are some slight symptoms at hop 2 (which is the border between our internal network and rr.com - so that's the cable modem), but starting at hop 6 there's some real bumps in latency and the jitter (latency variation) is also a big cause for concern.

In this case, both problems are inside the rr.com network, and since we are an rr.com customer, we would call them for help on this.

## **Example: 802.11b network near its range limit**



Here's an example where we're connected using a computer-based VoIP service (like Skype). Our computer is hooked up to our DSL modem via a wireless 802.11b network. Hop 1 is our DSL modem.

Here, we see a little bit of packet loss being added to every hop - our wireless network is losing a few packets (about 1 to 2 percent, it looks like), and this impacts everything this computer does - including our VoIP connection.

The call quality would be generally good here (probably better than acceptable - up to the "good" range, really). The latency is fine and there is very little jitter, but there is a little packet loss. There is a problem, though - at 9:31am, our call was interrupted - it looks like hop 1 lost a bunch of packets all together and during that period, we were unable to hear anything. Let's zoom in on that a bit.



See the period where we start getting a lot more packet loss, and then all the hops show a big block of lost packets - a period where it's likely no packets were getting through.

Here, the solution might be to move the wireless access point, or switch to wired on that computer.

## Reporting problems, when you find them

If you're using PingPlotter, it's almost certainly because you're experiencing some kind of problem - and when you find something that you think might be the cause of that problem, you need to communicate that to the right party.

We [cover this topic](#) in some depth in our Getting Started Guide. The piece we want to stress here is that the data in PingPlotter doesn't really mean anything unless you correlate it with a network problem (like poor VoIP quality). It's of paramount importance that your complaints include a description of how this problem is affecting you. Don't just send a graph from PingPlotter expecting them to be able to figure out what was wrong.

One great way of doing this is to put comments in the PingPlotter graph itself using the "Create Comment" feature of the time graphs. Make comments every time your VoIP quality is bad. Make comments when you give up on a conversation because they can't hear you at all (but you can hear them just fine - how frustrating!).

## Troubleshooting PingPlotter

**If you get "Destination Unreachable" at something beyond hop 3 or so, but can access that site via a web browser.**

Some sites do not respond to ICMP echo requests. See our knowledge base article for [instructions on how to configure PingPlotter to use TCP packets](#) instead of ICMP.

### If you get "Destination Unreachable" at hop 1 for all targets

Make sure your software firewall (ie: ZoneAlarm, etc) is configured to allow PingPlotter to have access to the network.

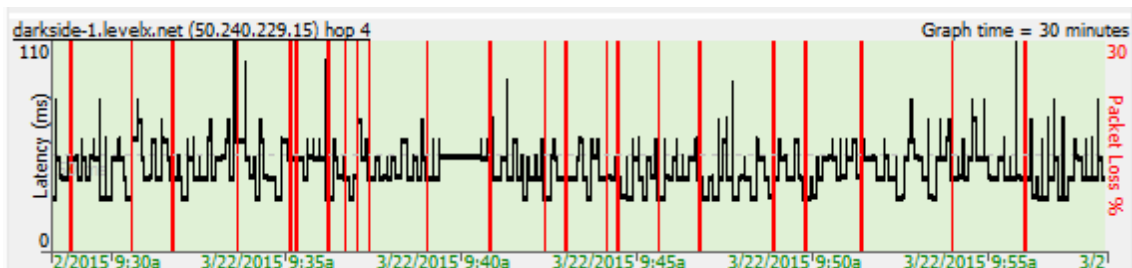
### If you only have the final hop visible - and all intermediate hops are empty

We cover this in this [knowledgebase article](#).

## Other questions

### Jitter

Jitter is the amount of variation in latency. If one packet takes 100ms and the next one takes 200ms, there's 100ms of jitter there. PingPlotter Pro offers jitter calculations and graphs, but PingPlotter Standard (and the 2.x line) still gives you an easy way to see the jitter by looking at the smoothness of the time graph, zoomed in a little. Here's an example:



This is zoomed in enough for us to see the individual samples, and we can see that none of them come in with the same latency. Adjacent samples here often have latency variations of 100ms, and just about every one has latency variation of 30ms or higher. Just looking at this graph, we can see a lot of jitter. Compare that to the first picture in this article - where the line was completely flat. We're looking for the flat lines, not big variations with red stuck in everywhere.

### Other resources

This article introduces some concepts and ideas about VoIP troubleshooting. There are other resources online that provide more depth (albeit not within the context of PingPlotter).

[www.whichvoip.com/voip-troubleshooting.htm](http://www.whichvoip.com/voip-troubleshooting.htm) - a great page with real-world solutions and suggestions.

Targeted to residential, but useful everywhere.

[www.voiptroubleshooter.com](http://www.voiptroubleshooter.com) is a great site that has an enormous amount of content on which symptoms relate to what kinds of network problems. This site has a relationship with [Telchemy](#), a leading VoIP provider of call quality monitoring tools.



## 6.11 Building a Compelling Case

If you find a network problem, you'll most likely want to try and solve that problem. In some cases you might be able to solve this network problem yourself - possibly by upgrading the BIOS on a hardware device, replacing a network cable or changing network service providers. In most cases though, the network problem will need to be fixed by someone else. If this happens, you'll want to build a compelling case that clearly demonstrates the problem, and then also convey (or present) that information. PingPlotter can help you do both of these tasks. It does this by collecting data over time, and then giving you the capability to present it in a way that can be compelling to someone else.

In this section, we'll show you how to use some of the tools in PingPlotter to collect data, save and reload that data, and then cover areas to focus on to make a really convincing story.

A convincing story (when using PingPlotter to document a network problem) is one that has some (or most) of the following attributes:

- Correlates the problem description (i.e.: bad VoIP quality or slow game performance) with the supporting data (i.e.: packet loss in PingPlotter).
- Is clear about the actual problem being experienced. Bad PingPlotter data is meaningless unless there is some impact to other applications. Make sure you describe the affect this problem is having on your network experience.
- Includes solid supporting data, covering at least a few minutes and possibly several days
- Does not exaggerate the problem or the data. You can certainly zoom PingPlotter in to only the very worst 10 samples you collected, but that doesn't give a realistic picture of the problem
- Is concise and not argumentative.

Most of the time your first contact with your provider (the one who can help you solve the problem) is going to be with a first-level, front-line support technician. In many cases, their goal is to "close the case". You need to be polite but persistent, and have a strong story to get them to bring the problem to the next level.

Because many ISPs and providers don't give copies of PingPlotter to their front-line support technicians, you probably won't be able to just send a PingPlotter save file for them to analyze. You'll need to create an image they can easily see in their email, or attach to their support case.

### Creating a graphical image of the problem

The main graphs in PingPlotter can be a compelling picture of the problem. In a lot of cases, you'll want

to include a snapshot of this screen in an email. There are a few ways to do this.

- Depending on your email software, you might find it easiest to just paste an image into an email or document. To do this, use the "[Copy as Image" option under the Edit menu](#)". This will put the image into the clipboard, and you can then paste it into your email or other document.
- You might, instead, prefer to create an image file and then attach it to an email. To do this, use the "[Save Image..." option under the File menu](#)". This will allow you to save an image in as a Windows bitmap, GIF or PNG file. Out of the box, the default file type is PNG, which creates the smallest possible file (with the highest quality as well!).

**Part**

---

**VII**

**Advanced Features**

## 7 Advanced Features

### 7.1 IPv6 and IPv4

PingPlotter will automatically pick the right address type when you enter a name. It uses the Windows-standard name lookup utility, so it will pick an IPv6 address over IPv4 based on how Windows works. Some targets (a growing number of them, "www.google.com" is one example) will let you use their services either way.

If you know specifically which protocol you want to use, you can prefix the name with the IP type, like this:

ipv4:www.google.com

or

ipv6:www.google.com

If your computer supports that IP type, then PingPlotter will use it. If not, you'll get an error.

### 7.2 Raw TCP sockets and WinPcap

TCP traceroute allows access to some targets that were not previously available, but this comes with some caveats.

One challenge that many users will face is that most operating systems today (all Windows versions XP or newer) block the ability to create TCP packets with the options needed for traceroute. See [a Microsoft white paper](#) for some more details on this.

The best way to get around this problem is to use a custom network driver. PingPlotter supports the use of [WinPcap](#) (and spinoff/clone variants) for this.

#### How to install the driver

Install steps:

- Visit the [WinPcap site](#) so you know what you're installing.
- [Download the driver](#). We recommend using the latest version. If you're using Windows Vista x64, then 4.0 or higher is required.
- Install the driver. The [WinPcap download page](#) has reasonably good instructions. Basically, run the installer.

- Use the TCP packet type in PingPlotter.
- You're done!

## Shortcomings with WinPcap and PingPlotter

WinPcap allows PingPlotter to send packets directly to the network card. This is very powerful, but also means that the Windows protocol stacks don't help us with routing or validation of the packet.

PingPlotter should do a good job with detecting if a gateway should be used and sending to that gateway, but might not work correctly with multiple gateways, or other complicated network setups.

Please [contact us](#) if you find a network environment where PingPlotter seems to be making poor decisions or causing problems.

If you're not running as administrator (or using PingPlotter as a service), then you'll need to configure WinPcap to start automatically (which is its default install mode). For more details on how to do this, see the WinPcap FAQ, [question 7](#) and [question 18](#).

## 7.3 Tracing from a remote server

PingPlotter is able to initiate trace requests from a remote agent on both Windows and UNIX based machines. The UNIX agent requires perl, a web server, and the traceroute command. The Windows agent is fully self-contained (it has its own TCP server and GUI component) and does not require any additional software or server components.

The basic concept of the remote trace agent is that it resides on a remote machine listening for a request for data from your local PingPlotter instance. When it receives a request, it does a single traceroute and returns the results back to PingPlotter, which then collects this data and displays it locally like any other trace data. The interval at which PingPlotter contacts the remote agent is controlled by the sampling parameters (trace interval) you've defined locally.

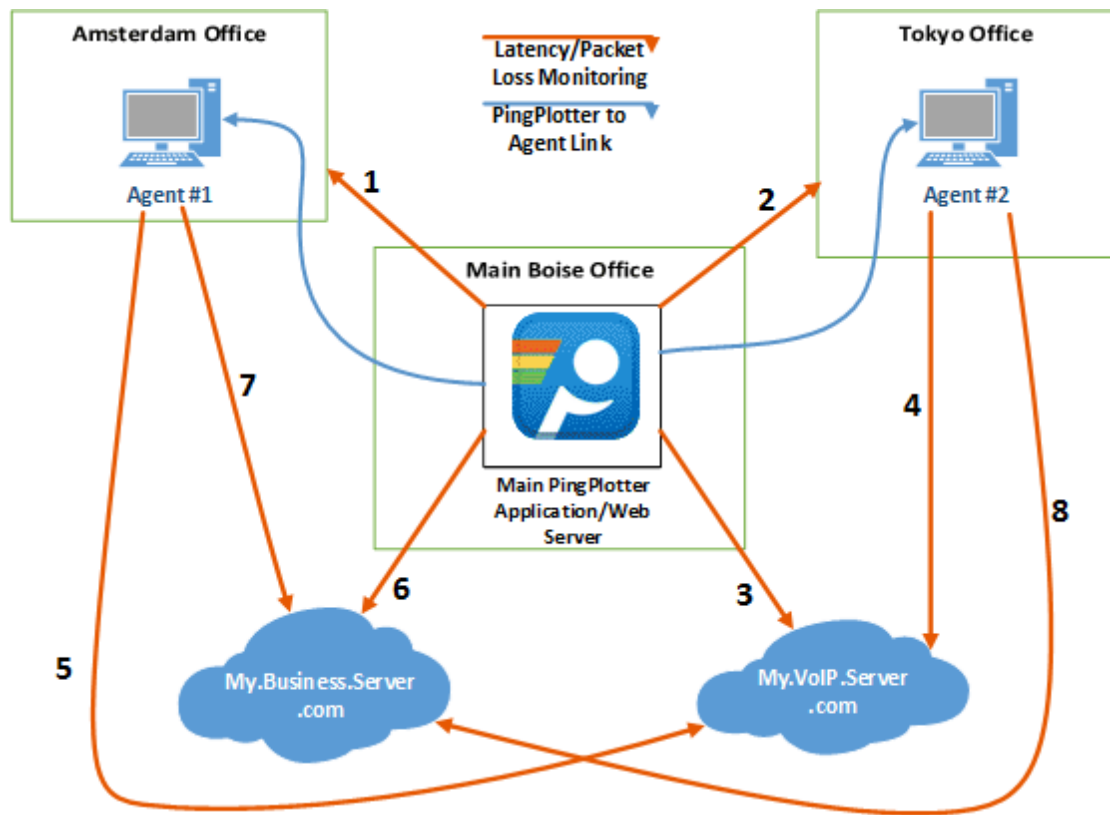
When using a remote agent, PingPlotter works and looks *\*exactly\** the same as it does when you're using the local engine. Essentially the only difference is that the trace isn't originating from the workstation running PingPlotter, but instead the agent is allowing you to source and see the traceroute from the vantage point of the machine running the remote agent. You can even see the results from multiple remote sources/agents by defining and using [Named Configurations](#) .

### Example:

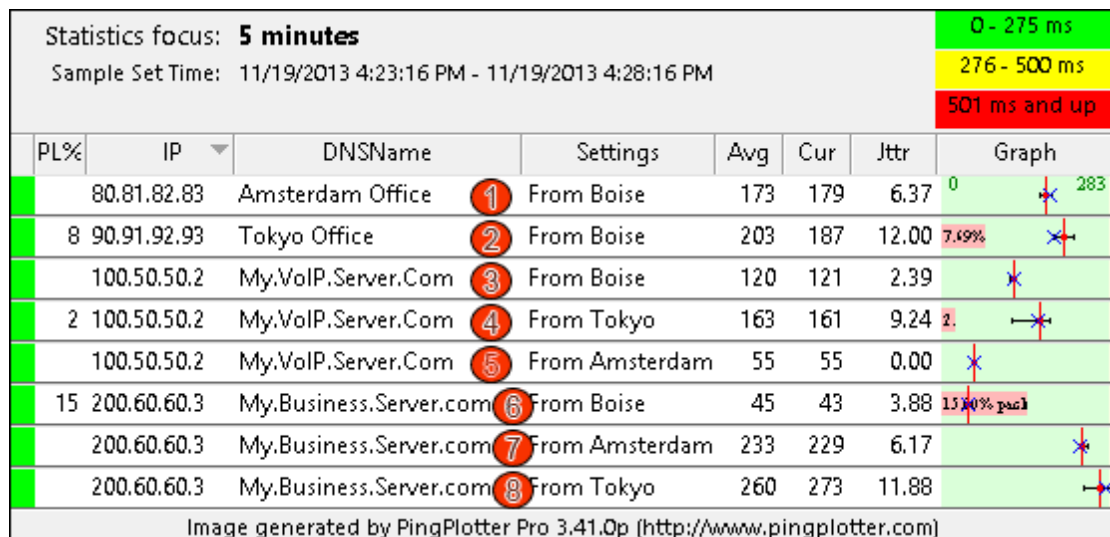
Lets say that we have three offices: a "main" office in Boise, a branch in Amsterdam, and one in Tokyo.

With a single instance of PingPlotter running at our Boise office, we already know that we can monitor the connection between us, and the other two offices. We can also monitor our connection from the Boise office to our business and VoIP server.

When using remote agents deployed on machines in both the Amsterdam and Tokyo office, we can now also measure the connection from both of those offices to both the business server, and VoIP server as well - all from the single instance of PingPlotter that's running back at our headquarters in Boise!



Our summary screen in PingPlotter would look something like this:



## Requirements:

- PingPlotter version 3.10.0p or higher, installed.
- A Windows XP or higher workstation (Windows agent), or a UNIX machine with Apache, perl, and appropriate permissions (UNIX agent).

## Installation instructions

### Installing the Windows Agent

The PingPlotter remote agent for Windows is a self-contained executable that includes a small server and some portions of the PingPlotter core trace engine. This agent can run as an application or as a service under Windows XP or newer.

Note: Presently the Windows agent does not come packaged with an installer or uninstaller.

- Download the Windows remote agent from [the agent download page](#).
- Extract the agent (PingPlotter\_Remote\_Agent.exe) to a working directory (e.g., c:\Program Files \PingPlotter Remote Agent).
- Browse to the directory where you extracted the agent and launch it. The agent will show up in the system tray.

- Ensure your firewall(s) allow access to the agent on TCP port 7465, or change the server port for the agent. To change the port, as well as other settings for the agent, double-click on the tray icon.
- From your PingPlotter machine, launch your web browser and access the agent at `http://(your agent machine name):7465`.
- Verify the output. It should look something like this (in your browser):

```
PingPlotter Pro Remote Windows Agent V0.8.2.18
Tracing to:
```

---

```
Error: Must specify an IP Address
```

Continue on to the [Configuring PingPlotter to use a remote agent section](#).

The windows agent has a few options, to see about its use, see the [configuring the Windows agent](#) section.

## Installing the UNIX Agent

The PingPlotter remote agent for UNIX is a perl or php script that uses the UNIX traceroute command to gather sample set(s) and pass the results to PingPlotter via a web server. Because of all the various flavors of UNIX that exist, and the possible differences when it comes to output of the traceroute command, you may need to modify the script for your particular source machine. We've primarily tested under FreeBSD and Debian. [Contact us if you have problems here](#).

- Download the unix agent script from [the agent download page](#).
- Extract the perl script (trace.pl) or php script (trace.php) into a web-accessible directory on a UNIX web server that is capable of serving the appropriate script.
- If necessary, set the script to be executable by your web server, (chmod 755, or similar, and don't forget to check the permissions for the directory).
- Test it from a web browser to make sure it's accessible. Example: [http://www.yourwebserver.com/pp\\_pro\\_agent/trace.pl](http://www.yourwebserver.com/pp_pro_agent/trace.pl) (or trace.php if you're using the php version).
- Test a full trace from the web browser. This requires the ?IP= parameter (?ip=parameter will fail, the 'IP' needs to be capitalized). This would look like this to trace to 192.168.1.1: `http://www.yourserver.com/pp_pro_agent/trace.pl?IP=192.168.1.1` Ideally you'll want to use a valid IP address here, of course - one that works on the remote agent's network.



- Verify the output. If it's working, you should get output that looks something like this:

```
PingPlotter remote trace agent, V0.4
```

```
1 192.168.1.1 0.684 ms
```

You'll probably have more hops than this. This is just an example if you were using an IP local to your server.

- If you have an error, look at the source code for the script for some different alternatives and help.

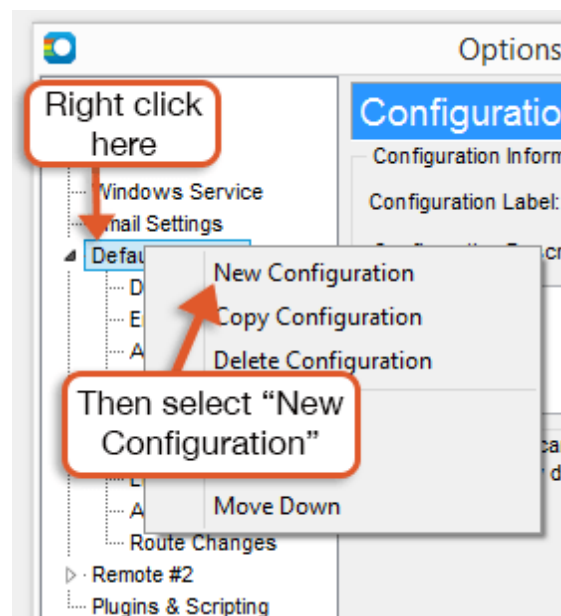
## Configuring PingPlotter to use a remote agent

Once the remote server(s) are configured, we need to set up PingPlotter so it knows about those servers.

### Creating a named setting (configuration) to use a remote agent.

Named configurations are covered in more depth in the [named configuration section](#).

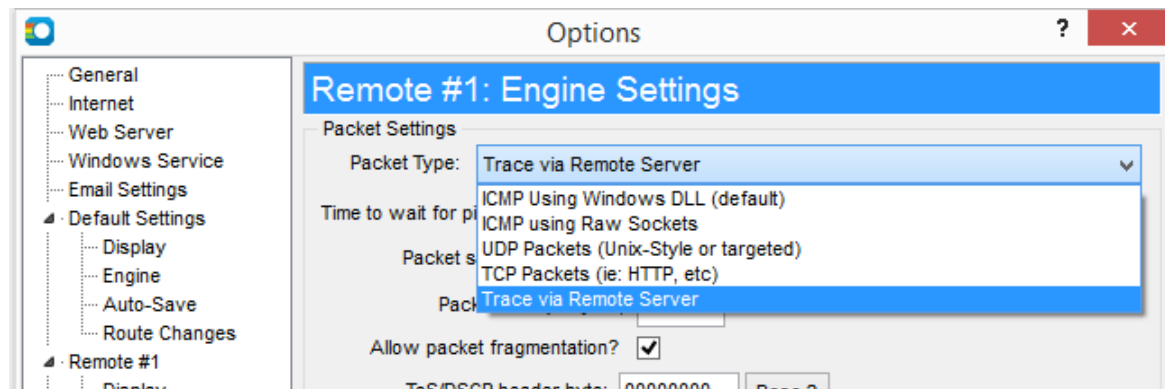
- Go back to the options dialog (Edit -> Options).
- Create a new configuration (Right-click on the "Default Settings" in the tree view and select "New Configuration").



- In the newly created configuration, change the name in the main section (e.g., "via <your server

name>").

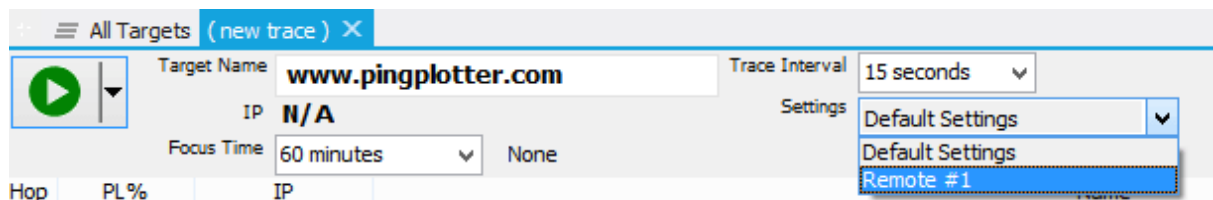
- Go to the "Engine" section of this new configuration and select "Trace via Remote Server" from the dropdown. (If this engine type is not listed, look at the [turning on the remote trace capability](#) section to enable the remote trace engine.)
- In the new section "Remote Server Source Settings", Enter the URL to the trace agent there. For a Windows based agent, the URL will look like this: **http://yourservername:7465/**. For a UNIX-based server, the URL will look something like this: **http://yourservername/script\_path/trace.pl** (or trace.php, if you're using the php version).



- Hit "OK" to close the options dialog and save your settings.

## Usage Instructions

Now, on the main PingPlotter screen you'll see a "Settings" dropdown below the "Address to trace" field. Pick "via (servername) ..." there (or whatever you named your configuration). Enter your target, and then click on the Trace button.



That should access your remote server and populate data into PingPlotter from that.

Warning: Your tendency will be to pick a server very familiar to you, which in a lot of cases is going to be the same server you just set up the script on. If you do this, your trace will only be one hop long!

## Error handling and troubleshooting

A remote agent adds additional complexity to any deployment, and PingPlotter's is no exception to this. In an attempt to keep things as simple as possible, and still satisfy our needs, we use existing technologies - web servers, http, existing command-line tools (for UNIX, where good ones exist), etc.

This means that in most cases we're dealing with familiar territory when troubleshooting. All the remote agent capabilities can be accessed via a web browser. If PingPlotter is having problems reaching a remote agent, then you can use your web browser to check and verify. You can also use PingPlotter (not the remote agent) to constantly monitor the availability of the server and route to the remote agent.

If the remote agent is inaccessible for some reason, there is a network failure between the PingPlotter machine and the agent for example, any alerts you have configured will not fire. If you want an alert to fire, you need to set up PingPlotter to monitor the agent machine itself.

There are a number of configuration challenges that might hit you here, but these should be familiar to you if you've done much network and/or application troubleshooting. ZoneAlarm, for example, might block access between PingPlotter and the agent machine. Other firewall issues could also come up. If you view the remote agent as a web server, the troubleshooting techniques you've used elsewhere should help you here too.

If PingPlotter is unable to reach the agent, because of an authentication problem or network failure, the error will appear above the trace window on the targets it applies to. This message might be an HTML error (e.g., 401 authentication failed), or a network failure message. These events will show up in PingPlotter's time graph the same way a route change does, with the background color on the time graph matching background colors elsewhere rather than having the graph scale colors painted on.

## Notes, thoughts, shortcomings...

The Windows agent uses a fully threaded engine to make working with it just as responsive as working with the local engine (if there is a timeout, other hops will still report back). For the UNIX agent, though, if hop 1 times out hop 2 won't report back until the timeout period has elapsed. This means there may be times, when the trace interval is < 5 seconds, where you see empty spots in the data that later fill in with a lost packet or a good sample. Everything works, just the results get delayed a bit. The Windows agent shouldn't have this problem.

## Configuring the Windows agent

Because the Windows agent is a full-fledged server unto itself, you may want to have it behave differently depending on where you're installing it.

### Server Port

The default server port of the Windows agent is 7465. If this doesn't work for some reason in your network, you can change it by right-clicking on the tray icon and selecting the "Settings..." option. Clicking on the "Apply" button makes that change immediately, and if the server is unable to start on that port an error will occur.

## Security

You might not want to expose the agent to anyone or limit access to it. This can be done via the username / password settings in both the agent and the remote trace setup in PingPlotter. The authentication system uses standard HTTP basic authentication. If the agent requires a username / password, and it doesn't match what's specified in PingPlotter, an error will show up in PingPlotter. If no username or password is required by the agent, then the username / password settings in PingPlotter will be ignored.

## Running the agent as a service

If you want to be able to trace from a machine at any time, you can install the agent as a Windows Service.

To install as a service, browse to the agent's install directory using a command prompt and then enter the following command:

```
PingPlotter_Remote_Agent /install
```

After doing this the agent will show up in the control panel services applet, or you can start the service manually with this command: **net start PP\_RemoteAgent**

To uninstall the service (remember that because there is no installer / uninstaller for now, you'll need to manually remove it), stop the service, then run this command:

```
PingPlotter_Remote_Agent /uninstall
```

You can also use the "/reinstall" parameter to register the agent in a new location.

If any of these commands have an error, then you'll get a popup message telling you about that error. If you have no errors, then the command will complete with no popup messages. If you do get an error, you can check in the "PP\_Remote\_Agent\_error.log" file to see what the error was.

Note that under Windows Vista and newer, you will not have access to the user interface when running as a service. For best results, stop the agent, run it as an application to change its settings and do testing. When you have everything working the way you expect, close the application and restart the service.

## Configuring the UNIX agent

The UNIX agent is much simpler (more primitive?). It's a perl or php script that runs the traceroute command and shows the output on a web client. Because it uses an existing web server, all the settings need to be configured on that server including the server port and security.

## Security

Because PingPlotter uses standard basic authentication to secure access to the agent, you just need to password protect the agent application using an .htaccess security setting (if you're using apache, which most people are). PingPlotter will send the username and password to the web server using this protocol, and it should all just work.

As always, please [contact us with questions](#).

*\*\*The feature in this topic is exclusive to PingPlotter Pro. See our [product comparison](#) page for more details\*\**

## 7.4 Command Line Arguments

You can have PingPlotter do a few things automatically on startup by specifying command line parameters. You can put these parameters in a shortcut - or enter them from a DOS command line window.

```
PingPlotter.exe [File to Load] [/TRACE:[Address To Trace]]
```

### Loading a file at startup

If you enter a parameter without a / on it, PingPlotter will try and find a file by this name - and load it if it's found.

Example:

```
PingPlotter.exe www.pingplotter.com.pp2
```

### /TRACE

This option will start tracing automatically when PingPlotter loads. If use the option to load a save file on startup, then tracing will begin to this address. Otherwise, Add a colon (:) and the IP Address or server name you want to trace to.

Example:

```
PingPlotter.exe www.pingplotter.com.pp2 /TRACE
```

or

```
PingPlotter.exe /TRACE:www.pingplotter.com
```

**/?**

Show this help screen.

Example:

```
PingPlotter.exe /?
```

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

## 7.5 Installer Options and MSI

The PingPlotter install is MSI-based, and wrapped with a bootstrap that helps do upgrades.

Normally, the best way to install PingPlotter is just to launch the installer by downloading and double-clicking.

In some cases, though (particularly for deployment to multiple computers), it may be helpful to change the way PingPlotter is installed.

The bootstrap has several options to extract the MSI, log the install to file, debug, etc. To see the options, launch the installer from a command line and pass it a `/?` parameter - this will list the parameters that can be used.

### License Entry

PingPlotter license are stored in the registry in the following location:

```
[HKEY_LOCAL_MACHINE\Software\Pingman Tools\PingPlotter\User]
```

```
"UserName"="Your Username"
```

```
"RegistrationCode"="Your License Key"
```

If you need to automate license key entry, you can write these values into the registry.

\*\*\* 64 bit windows warning \*\*\* - Since PingPlotter is 32 bit application, on a 64 bit machine, this needs to be written to:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Pingman Tools\PingPlotter\User]
```

### Custom Options

---

There are several other options available through the command line interface of the installer. Please [contact](#) our support team with your needs, and we'll help you build the right install package.

**Part**



Reference

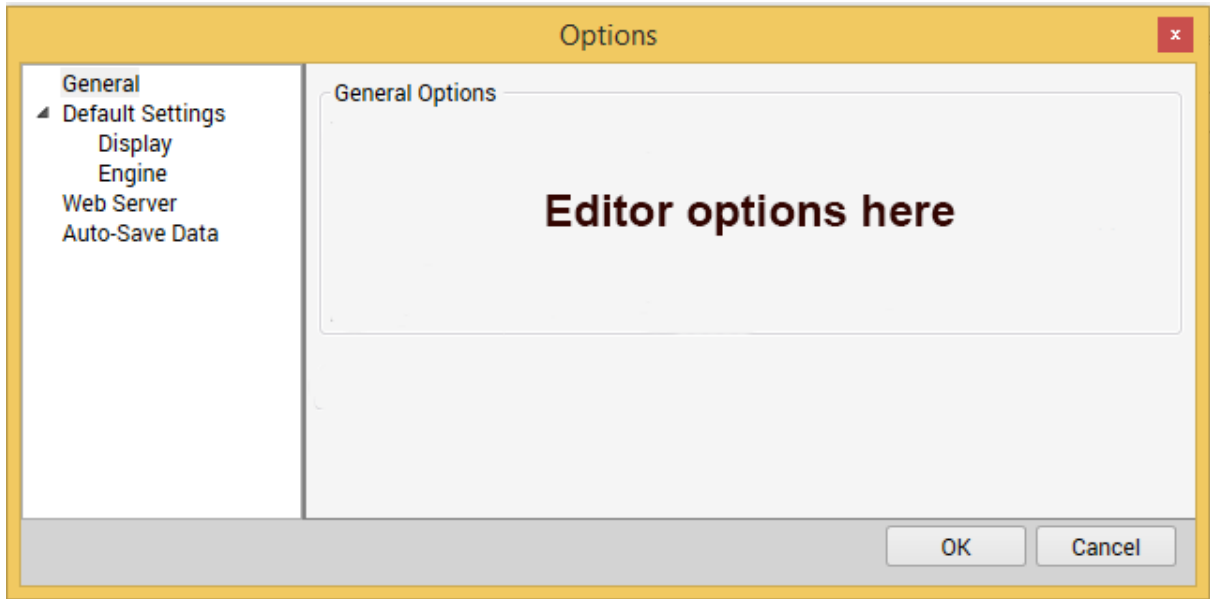


## 8 Reference

### 8.1 Options

#### Applying Pending Changes

Any changes made to these settings do not take affect until you either hit the "OK" button.



Hitting cancel will back out the changes that have not been applied. You can make changes to multiple areas before applying them.

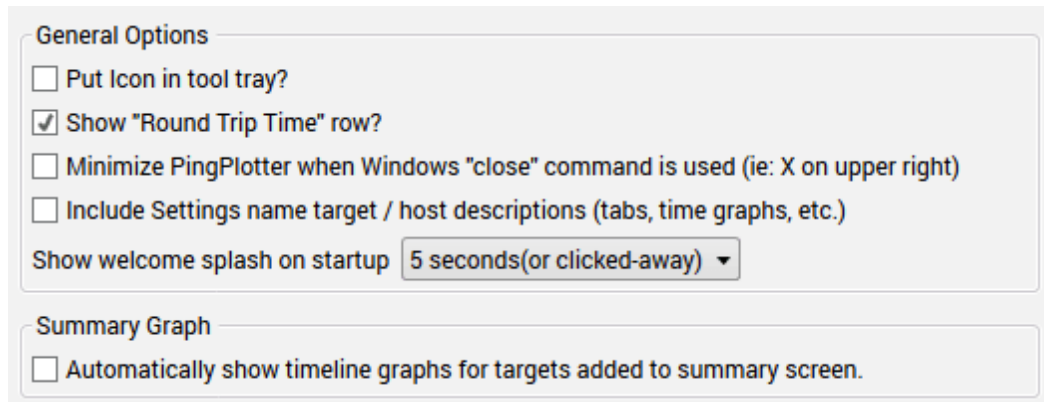
#### Creating / Manipulating named configurations

PingPlotter Pro supports [multiple named configurations](#) . Right clicking the "Default Settings" option above will allow you to create more.

If you delete a named configuration that is currently in use by a target, that target will automatically change to use the first configuration in the list.

***\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\****

### 8.1.1 General Options

The image shows a screenshot of the 'General Options' dialog box in PingPlotter. The dialog has a title bar 'General Options' and a close button. It contains several settings: 'Put Icon in tool tray?' with an unchecked checkbox; 'Show "Round Trip Time" row?' with a checked checkbox; 'Minimize PingPlotter when Windows "close" command is used (ie: X on upper right)' with an unchecked checkbox; 'Include Settings name target / host descriptions (tabs, time graphs, etc.)' with an unchecked checkbox; and 'Show welcome splash on startup' with a dropdown menu set to '5 seconds(or clicked-away)'. Below these is a section titled 'Summary Graph' with a checkbox 'Automatically show timeline graphs for targets added to summary screen.' which is unchecked.

#### Put icon in tool tray?

PingPlotter can be minimized to the "tool tray", the small icon "tray" where the clock normally sits, and where a number of other notification icons might appear.


Enabling this option will turn on a tray icon at all times. When PingPlotter is minimized, it will only be visible in the tray. When it's not minimized, it will show on the taskbar and the tool tray. Alert conditions can be surfaced through the tool tray as well, in which case the icon will change to red and a message might appear.

#### Show "Round Trip Time" row?

The "Round Trip" row of PingPlotter duplicates the information from the final hop and makes it evident that the host is reachable and what the round trip time and latency is. Before adding this option, we got a lot of questions "What's the round trip time?"

#### Minimize PingPlotter when Windows "close" command is used

Turn on this option to cause PingPlotter to minimize instead of close when the "X" button is hit.

If you normally run PingPlotter all the time, you might not want it to close if you accidentally hit the "close" button on the application (ie: the  button). Turning on this option will make PingPlotter minimize instead of close. To close PingPotter, use the "File" -> "Exit" menu option, or if PingPlotter is minimized to the tray, use the right-click "Close" command. Note: using the close command from the taskbar will \*not\* close PingPlotter, as that is equivalent to using the X button.

#### Include settings name in target / host descriptions

In PingPlotter Pro, when tracing to the same target with different engine settings (see our [named configurations](#) documentation for more details), the only way to distinguish between the settings is by the named configuration. If you're only using one named configuration (or if you're tracing to all different

targets), then this is not so helpful. Turn on this option to show the named configuration on all tabs and time graphs. For the summary graph, turn on the "Settings" column to show the named configuration used for that target.

### Show Welcome Splash on Startup

This option allows you to adjust the amount of time the splash screen is shown on start up of the program (there's also an option in the drop-down to not show the splash screen at all).

### Summary Graph Settings

[Summary graphs](#) are exclusive to PingPlotter Pro.

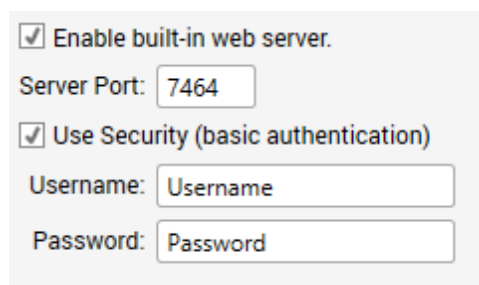
### Automatically show timeline graphs for targets added to summary screen

If you regularly trace to a lot of targets (and have them auto-show on the summary screen), turn this option off to control visibility manually.

When a new target or router is added to the summary screen, having a time graph automatically show up can be handy. The downside of this is that a long list of targets quickly fills up the screen and becomes less useful. Turn off this option if you find yourself regularly turning off a lot of the automatically added graphs. You can turn them back on at any time manually anyway.

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

## 8.1.2 Web Server Options

A screenshot of the 'Web Server Options' configuration window in PingPlotter Pro. It features a light gray background with several settings. At the top, there is a checked checkbox labeled 'Enable built-in web server.' Below this is a 'Server Port' label followed by a text input field containing the number '7464'. Further down is another checked checkbox labeled 'Use Security (basic authentication)'. Below this checkbox are two text input fields: 'Username' with the placeholder text 'Username' and 'Password' with the placeholder text 'Password'.

The PingPlotter Pro integrated web server allows access to PingPlotter from a remote machine. The user interface and functionality of this web server is designed to give the owner of this machine access to trace information remotely.

### Enable built-in web server

The built in web server will automatically serve a web-based user interface for PingPlotter, when enabled. This option is disabled by default (for security reasons), but can be easily enabled. This uses the

PingPlotter built-in web server engine, which is loosely similar to IIS, but not exactly. The supplied web interface is written in ASP, using VBScript. You can, however, use IIS as the web server and point at PingPlotter "www" directory and use IIS instead. If you are going to customize the scripts, it is recommended that you use IIS instead of the PingPlotter built in web server.

When enabled, you can access PingPlotter's web interface through this URL:

<http://localhost:7464/>

### **Server Port**

The web server port controls which port you would use to access the built in web server. If you use the default of 7464 (or PING on a telephone keypad), then you'd access the PingPlotter web interface through the above URL. Some other value would require a change in the URL.

### **Web server security settings**

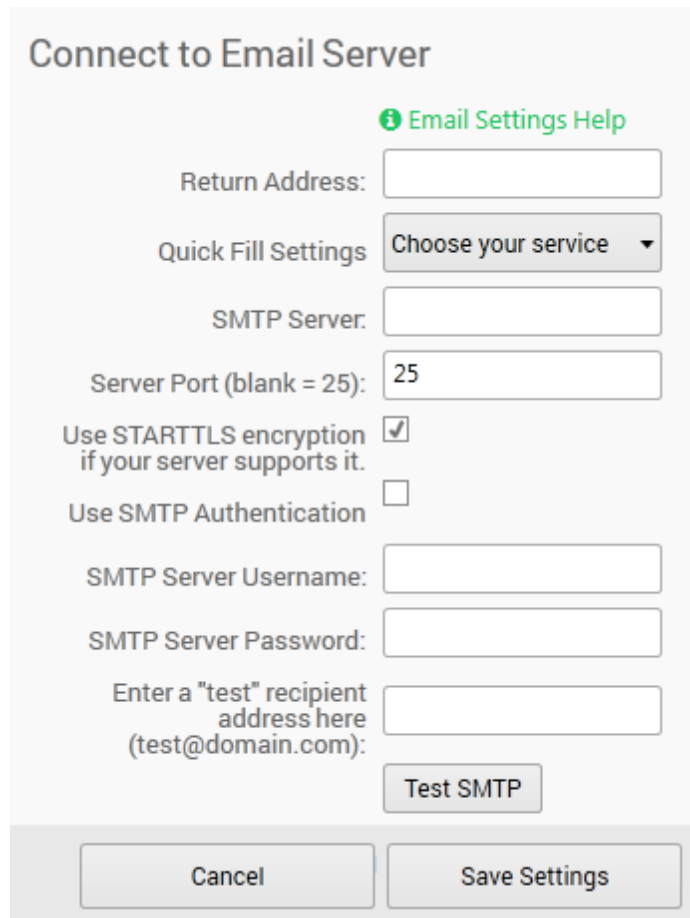
Enable this option to have your browser prompt for a username and password before displaying anything.

For best security, you'll want to require a username and passport to access PingPlotter's web interface. If you turn off security, then anyone with a browser who has network connectivity to your machine will be able to manipulate your PingPlotter sessions (including adding new targets and closing existing targets).

*\*\*The feature in this topic is exclusive to PingPlotter Pro. See our [product comparison](#) page for more details\*\**

## **8.1.3 Email Settings**

The email setup dialog is used to set up emailing for alerts. If you're not using alerts, or you're not interested in having the alert system email you, then setting this up is not required.



The image shows a web form titled "Connect to Email Server". At the top right of the form area is a green link "Email Settings Help" with an information icon. The form contains several input fields and checkboxes. The "Return Address" field is empty. The "Quick Fill Settings" dropdown menu is set to "Choose your service". The "SMTP Server" field is empty. The "Server Port (blank = 25)" field contains the number "25". The "Use STARTTLS encryption if your server supports it." checkbox is checked. The "Use SMTP Authentication" checkbox is unchecked. The "SMTP Server Username" field is empty. The "SMTP Server Password" field is empty. Below these is a field for a "test" recipient address, with the placeholder text "(test@domain.com)". A "Test SMTP" button is located below the test address field. At the bottom of the form are two large buttons: "Cancel" and "Save Settings".

### Return Address

All outgoing emails will have a return address specified, and this is the address that is used. Please make sure you specify a valid address here since this is where all the bounce messages will come from. Some ISP SMTP servers only allow emails sent out with a "from" address of their domain as well, so if you're having problems getting the SMTP server to work, make sure you're using a valid return address.

### Quick Fill Settings

This dropdown will auto fill in the SMTP server and port information for a few more common email providers (Gmail, Yahoo, Outlook, Office 365, etc).

### SMTP Server

The SMTP server is the server that your outgoing mails will go through. This may have been given to you by your ISP or your mail administrator.

### Server Port

The default port for most SMTP servers is 25. If you connect to your SMTP server via a different port,

then enter that port here. Leaving this blank will use port 25. If you're using STARTTLS/SSL, then this might be port 587 or some other port as supplied by your email server/provider.

### SMTP Authentication

Some SMTP servers require a username and password to be able to deliver mail. If this is the case with your server, turn on the "Use SMTP Authentication" checkbox, and then enter your username and password. The password is saved in your PingPlotter.ini file using a basic XOR encryption scheme – this will keep your password hidden, but this encryption method is "crackable" if someone really wants to figure it out by looking at your .ini file and reverse engineering it.

### Using STARTTLS encryption

PingPlotter also supports use of SSL (STARTTLS) for SMTP. For more details on this, [see our web site](#).

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

## 8.1.4 Display Options

Warning (yellow) speed  ms

Critical (red) speed  ms

☒ Draw line to show Min/Max range

Graph Scale

☒ Automatically scale to last visible time graph's sample times

☐ Automatically scale to individual graph sample times

☐ Scale to:  ms

Packet Loss

☐ Don't graph packet loss

☒ Packet loss Scale at %

Jitter

☒ Add jitter graph above time graphs?

Graph Height  points

Hide if area shorter than  points

☐ Automatically scale to individual jitter values

☒ Scale to:  ms

The "Display" settings control the general display format of PingPlotter's graphs, including scaling, coloring, and other general values.

### **Warning and Critical speed limits**

These boxes control the point at which the colors change. By default, all response 200 ms and below will paint green. From 201 to 500 will paint yellow, and over 500 will paint red. These numbers apply to both the HOP column and the graph background. In addition, the legend on the graph screen will be updated with these number.

You'll probably want to change the numbers based on your internet connection speed. If you've got a T1 or a cable modem, the listed numbers are probably pretty good (you might move them down a little if you're tracing to a fast site). If you have a modem, you probably want to crank these numbers up a bit.

Changing these values sets the Green / Yellow / Red threshold for the graphs. This is dependent on your expected performance. For a modem, 200 ms might be quite good, while for a T1, it could be considered bad.

### **Draw line to show Min/Max Range**

Show the Min/Max Line. Hide this line to keep the scale of the upper graph in better range. This line can be useful to understand how a specific hop is responding - for example, if hop 8's minimum point is significantly greater than hop 7's maximum point, then you may need to investigate what's happening between hops 7 and 8. It may be distance (ie: speed of light latency), or it may be a problem with one router, or the connection between those routers.

When showing just a few samples, this can be really handy to see the range of latencies. As you increase your window, though, a single bad sample can make this line stretch the scale of the graph.

### **Graph Scale**

The number shown here indicates the graph scale, in milliseconds (1/1000th of a second).

If the "Automatically scale to last visible time graph's sample times" option is being used, then this number is the maximum response time of **only** the final destination's sample set. This number can change (and WILL change) as new samples are received. The "Automatically scale to individual graph sample times" will also adjust the graphs automatically - but will scale each graph individually based on the maximum response time for that hop only.

If a fixed scale is being used, this number will always equal that scale. You can change to a fixed scale in the options screen.

### **Packet Loss**

The red number on the right of the timeline graph is the scale of the packet loss numbers. Depending on the number of samples included in the timeline graph, all timeouts may show 100%. For more details, [click here](#).

Most often, graphing the packet loss is a handy, easy way to see lost samples. 30% seems to work great for highlighting just the right of loss in most cases, but you're certainly going to run into cases where you want to change this to something lower (as low as 1) or higher (any number is valid – even over 100).

### **Jitter**

Jitter is a number that represents how stable the latency responses have been. A low jitter number is usually an indicator of a good connection. High jitter can lead to slow response times, poor voice quality (in Voice over IP) and other connection problems.

PingPlotter Pro allows you to graph jitter correlated with the time graph. In many cases, jitter is apparent when examining the standard PingPlotter time graphs, so the jitter graph is only displayed when there is enough room. The settings here allows you to control when that is displayed, and what it will look like.

### **Jitter Graph Scale and Target line**

The jitter graph scale controls the range of jitters you expect, and at what point the jitter will go off-scale. Normally, a jitter of much over 60 ms is indicative of a problem, so 60 is a good starting point. Any values over the scale will show with a red line for "overscale".

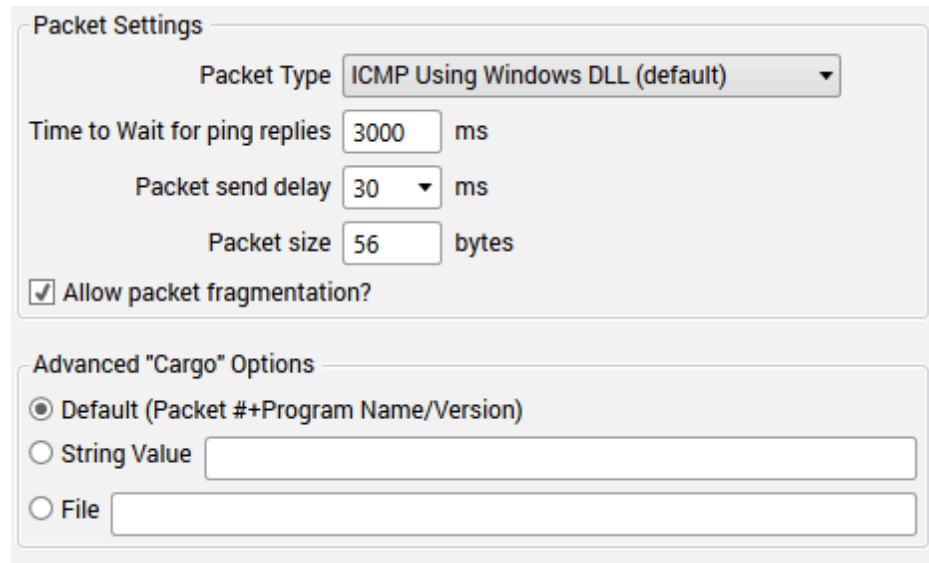
This graph scale is used in tandem with a "target line" which will be drawn across the graph at the point you specify. This is used to easily identify if jitter is exceeding your target number.

These settings are represented in milliseconds, and control the range of jitter values that will fit into the graph. Note that this also applies to the web interface, so if you want the jitter graphs to show up in the web interface, these settings need to fit the height of the web interface time graphs.

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**



## 8.1.5 Engine Options



**Packet Settings**

Packet Type: ICMP Using Windows DLL (default)

Time to Wait for ping replies: 3000 ms

Packet send delay: 30 ms

Packet size: 56 bytes

☒ Allow packet fragmentation?

**Advanced "Cargo" Options**

☒ Default (Packet #+Program Name/Version)

☐ String Value

☐ File

The "Engine" settings control what and how PingPlotter sends data.

### Packet Type

The "Packet Type" settings allows you to pick what kind of data you want to send to tailor PingPlotter to your network needs. PingPlotter supports 4 packet types:

- **ICMP using Windows DLL.** This method is the traditional method and matches the data that the Windows TRACERT command uses. It works on all Windows operating systems, and is a good balance of reliability and capability. This is reliable with the least CPU usage (on most operating systems). This method will automatically do manual timings on less-than-accurate operating systems to attempt to get accuracy to 1ms. This method does not require administrative rights, and should be the first choice for most users.
- **ICMP using Raw Sockets (advanced use only).** In some rare cases, the standard Windows method doesn't work. PingPlotter can compose its own ICMP packets - although in most cases this is no more reliable or better than ICMP.DLL. This requires administrative rights, and doesn't work reliably on Windows Vista or newer (including Windows 8).
- **UDP Packets (Unix-Style).** This uses ports 33434 – 33500 and closely mirrors Unix's traceroute command. This method will sometimes allow you to trace to a destination that isn't reachable via ICMP, or might allow you to reach the internet even if your ISP is blocking ICMP echo requests. Though not the cure-all for "Destination Unreachable" issues, this is worth a try, especially if you're getting erratic packet loss or unreachable destination. This requires administrative rights, and doesn't work reliably on Windows Vista or newer (including Windows 8).

- **TCP Packets.** This method gives you the opportunity to send TCP packets. If a firewall is blocking ICMP packets, it's sometimes possible to get a response using TCP packets instead. TCP is the protocol used for all web browsers in addition to FTP, Telnet and others. This requires Administrative user rights, and on most operating systems also requires a helper library.

### **Time to wait for ping replies**

This option allows you to fine-tune your performance a little. When PingPlotter sends out a packet, it waits a certain amount of time for a response. The longer it waits, the more resources it needs to use (to keep sockets open), but the more likely that it will get a response. By default, PingPlotter will wait for 3 seconds for any packet to return. If the packet doesn't return in 3 seconds, then it is counted as a lost packet. If patience isn't one of your virtues, you can turn this down somewhat. No matter what your value is here, timed out packet will show with the time "9999."

Because of the performance enhancements offered by PingPlotter, it's unlikely that this option needs to be changed. If it's set too low, it can cause misleading data to be generated.

### **Packet Send Delay**

This can be an interesting number to manipulate. It's really meant for "advanced" users, so you don't NEED to change it.

PingPlotter sends out multiple packets at the same time and times everything at once. Actually, it leaves a tiny interval between each packet so as not to completely saturate your bandwidth when it sends out 30 packets. This time interval is adjusted by this parameter. Most of the time, 25ms is good. This falls within realm of what a 28.8 modem can perform. If you've adjusted your packet size, or your connection to the internet is really slow, you might want to crank this number up a little. If you have just oodles of bandwidth, you can crank it down a little. Be aware that a too-small number can adversely affect your data.

### **Packet Size**

The Packet Size can make a considerable difference in latency performance. Normally, you want to use a relatively small number here. The default is 56 bytes, but in some cases you might need to lower this (especially on TCP port 80 packets, which sometimes get dropped unless they are 40 bytes). 1500 is lot of data, and should be used with great care. A 1500 byte packet means PingPlotter will be sending out 30-50 K per second worth of data, which can cause its own problems (and makes measuring latency more challenging).

### **ToS/DSCP header byte**

This is relatively advanced course material – ie: if you don't know what this means, just leave it at 0.

The DSCP byte is often used by network providers to determine packet priority and sometimes make other decisions about the data. There may be an occasion where you want to manipulate this byte and test network performance. VoIP data, for example, is often characterized by a value in the DSCP byte.

This byte can be edited as Decimal, Hex, or Binary by hitting the button beside the edit field. The display on the button shows the current format this value is displayed in.

Note that under Windows XP and newer operating systems, a registry setting needs to be modified to allow this byte to be used. This is a system-wide setting that allows applications to write their own values into this byte. If your system needs to have this value set, PingPlotter will prompt to see if you want this setting changed (and a reboot will be required) as soon as you modify the value in this field.

### **TCP Specific Settings**

When using TCP packets, you can specify which target port to use. Usually, you'll want to use port 80 here, but you're welcome to use any reasonable port. Windows firewall blocks creation of TCP packets, so you'll need to use WinPCap to create packets under that OS (and possibly others).

### **Cargo**

Note: This is an advanced option that should probably not be changed. This is used to diagnose network problems when specific \*data\* is sent - which is a highly unlikely problem for most networks.

By default, PingPlotter (basically) pads the outgoing packet with repeating string representing the current PingPlotter version and name. An example of this is PingPlotter250, repeated over and over to fill the cargo. If you suspect that your network may be having problems when you send specific byte codes, you can enter the hex code that you want repeated, OR a link to a file to read the byte string from. The cargo space for the packet will be padded with this data. Use this in conjunction with the packet size to create the network scenario you're looking to duplicate.

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

### 8.1.6 Auto-Save Options

The screenshot shows the 'Auto-Save Data' and 'Manage Data' sections of the PingPlotter settings. The 'Auto-Save Data' section has three radio button options: 'Always', 'After: 30 minutes' (which is selected), and 'Only when selected'. The 'Manage Data' section contains a text box for 'Data Storage Folder' with 'default' and 'browse' buttons next to it. A note below the text box states '(close/restart of client and service **required** after change)'. At the bottom of the 'Manage Data' section, there is a warning: 'It's **really** important that this folder is always accessible (and if you're running as a service, it needs to be accessible by the service). **This should be a folder on a local drive.**' A 'View/Manage Sessions' button is located to the right of the text box.

**Auto-Save Data**

Automatically keep collected data:

☐ Always

☒ After:

☐ Only when selected

**Manage Data**

Data is automatically stored as it is collected, continuously. Collected data is accessed via your running workspace, or via the session browser in the File menu. Read more in [the PingPlotter manual](#).

(close/restart of client and service **required** after change)

It's **really** important that this folder is always accessible (and if you're running as a service, it needs to be accessible by the service). **This should be a folder on a local drive.**

#### Auto-save data

There are a few different options for auto-saving data:

- "Always" - this will keep every target's data stored in PingPlotter's database
- "After:" - this setting will only keep sessions that have met or exceeded the timeframe that is entered
- "Only when selected" - this will result in the program not saving any data for a target unless you explicitly instruct it to do so

If the "Only when selected" option is chosen, the option to select a target to save data for can be accessed by opening a trace window, and right clicking on the target tab. Here we have options to "Discard on close," or "Keep (auto-save)." These options are covered in more detail in the [auto-saving of data](#) section of the manual

#### Manage Data

PingPlotter won't automatically delete any trace data if we stop tracing and close a target. The program will keep the data on file for a limited amount of time before it's deleted - so there's an opportunity to recover past trace sessions. The previous trace sessions can be found by clicking "View/Manage Sessions." From the session browser, there are options to reopen or delete a previous session, as well as the option to export a session (which will save off a .pp2 file).

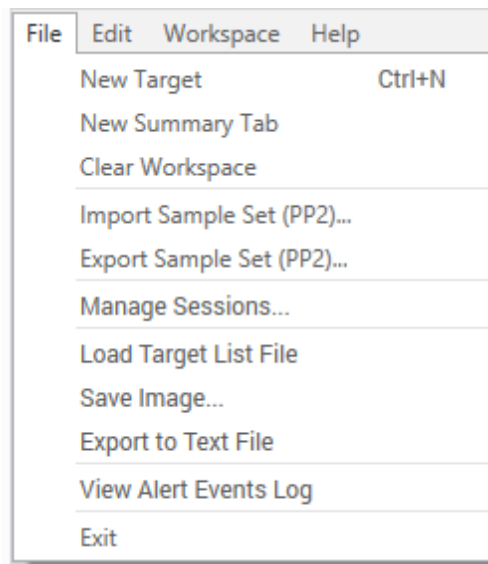
#### Data Storage Folder

All of the data in PingPlotter is saved to a file titled "sessions.ppdata" in one of two locations. If we're running PingPlotter as a Windows service the file's default location is in the C:\ProgramData\PingPlotter 5\ folder. If we're running the program as an application, then the file's default location is in our user application data folder C:\Users\\*\*username\*\*\AppData\Roaming where \*\*username\*\* is the username of the currently logged in user. The option here can be used to change the location where the sessions.ppdata is stored (and a restart of the program and/or service is required after changing this setting).

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

## 8.2 Menus

### 8.2.1 File Menu



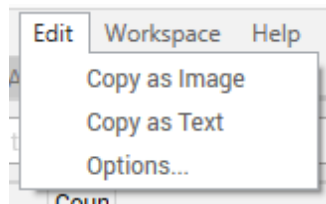
1. **New Target...** - This will create a new empty target area where you can trace to a new instance. See the documentation on "[Tracing to Multiple Targets](#)" for more details.
2. **New Summary Tab** - In PingPlotter Pro - this will create a new summary tab (we cover custom summary tabs in more detail [here](#) ).
3. **Clear Workspace** - In PingPlotter pro - this will clear out your current workspace
4. **Import Sample Set...** - Loads a previously saved sample set. The default extension for PingPlotter saved sample files is .pp2, or PingPlotter save file format.
5. **Export Sample set...** - Allows you to save the current sample set to an external file. These files are

saved in .pp2, or PingPlotter's save file format.

6. **Manage Sessions** - Opens the session browser, where you can open, export, or delete previous trace sessions
7. **Load Target List File** - Allows a list of targets to be loaded into PingPlotter
8. **Save Image..** - Saves the current graph in .png, .gif or .bmp format. See the [Autosave section](#) for information on how to automate the saving of graph images.
9. **Export to Text file...** - Exports trace data to a comma delimited text file. Click [here](#) for an explanation of the export options available from PingPlotter.
10. **View Alert Events Log** - Opens a log that displays information on when/why any alerts may have fired.
11. **Exit** - Exits PingPlotter. By default you'll be prompted to save your current sample set if you haven't done so already (click [here](#) to see how to change this option).

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

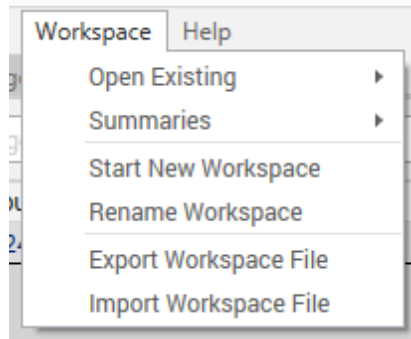
## 8.2.2 Edit Menu



1. **Copy as Image** - Copy the current graph to the clipboard as an image. From here, you can paste the image into your favorite graphics program or an email.
2. **Copy as Text** - Copy the current graph to the clipboard as text. Hold down the shift key when clicking the Edit menu to toggle between copying all the collected data details, or copying a summary.
3. **Options...** - Go to the configuration and options setup area.

*\*\*Some of the features listed in this topic are only available in PingPlotter Pro and/or PingPlotter Standard. See our [product comparison](#) page for more details\*\**

### 8.2.3 Workspace Menu



Workspaces are used in PingPlotter only, and are a list of targets, screen locations and settings that make it easy to continue a monitoring session later. We discuss this in some detail in "[Tracing to Multiple Targets](#)".

1. **Open Existing.** - This will load a workspace - which is a list of targets, trace intervals, [named configurations](#) and screen locations. PingPlotter workspaces use the extension **.pws** by default. Loading a workspace will stop tracing and close any targets that are currently active.
2. **Summaries** - Lists any summary screens that are present in the current workspace
3. **Start New Workspace** - Creates a fresh blank new workspace
4. **Rename Workspace** - Allows renaming of the current workspace (which can make it easier to pick workspaces out of the "Open Existing" list)
5. **Export Workspace File** - Exports the current workspace as a \*.pws file
6. **Import Workspace File** - Opens any \*.pws file that may have been saved/shared

*\*\*The feature in this topic is exclusive to PingPlotter Pro. See our [product comparison](#) page for more details\*\**

# Part

---



IX

More Information



## 9 More Information

### 9.1 Support

We offer a number of great support resources for PingPlotter that might be able to answer your questions without having to ask us. We're always happy to get an email from you as well though.

**Support Page:**

<http://www.pingplotter.com/support.html>

**Troubleshoot and Solve Your Network Problem:**

<http://www.pingplotter.com/netnirvana/>

**A Lineup of Common Network Problems:**

<http://www.pingplotter.com/commonnetworkproblems/>

**Getting Started Guide:**

<http://www.pingplotter.com/gsg/>

**Tutorial and Product Manual:**

<http://www.pingplotter.com/manual>

**Knowledge Base:**

<http://www.pingman.com/kb/>

**Support forums:**

<http://www.pingman.com/forums>

**Email Support:**

[support@pingplotter.com](mailto:support@pingplotter.com)

### 9.2 Purchasing

PingPlotter has three different editions - Free, Standard, and Pro. To see more info about each version, be sure to check out our [feature comparison page](#).

PingPlotter Free doesn't require any sort of license key - and you're welcome to run that version of the program for as long as you'd like at no cost.

PingPlotter Standard and Pro both require a purchased license key to run after their 30-day evaluation period has expired. If no license is entered after the evaluation period has expired, the program will revert to PingPlotter Free.

We offer a variety of ways to purchase licenses, including credit card, PayPal, check, money order, or via purchase order (after approval). The very easiest way to order is with a credit card online – you'll get a license key immediately, and you won't have to wait for your order to be processed by hand (as some other methods require).

To purchase a license, visit our online order page at:

<http://www.pingplotter.com/order.html>

Thank you!

# Index

## - A -

Agents 105  
Alerts Setup 31  
Alerts Setup: Event Notification 41  
Alerts Setup: Event - Log to file 44  
Alerts Setup: Event - Play a Sound 43  
Alerts Setup: Event - Send Email 43  
Alerts Setup: Event - Tray Icon Change 44  
Alerts Setup: Troubleshooting Alerts 42  
Applying changes 117  
Auto-save 128

## - B -

Browser 27

## - C -

Cargo 125  
Clipboard 57  
Closing Target 54  
Colors 122  
Column Widths 18  
Command Line Arguments 113

## - D -

DNS Name 17  
Docking 54

## - E -

Engine type 55  
Examples 89  
Exporting 57, 58

## - F -

Floating 54

## - H -

History Graph 18

## - I -

Image creation 128  
Introduction 2

## - J -

Jitter 89  
Jitter Display 122

## - L -

Latency 12  
License Key Entry 114

## - M -

Menu  
    Edit 130  
    File 129  
    Workspace 131  
Minimize instead of close 118  
Multiple Targets 54

## - N -

Named configurations 55

## - O -

Options 117, 122

## - P -

Packet size 125  
Packet type 125  
Ping Thresholds 122

## - R -

Registration 133  
Remote Tracing 105  
Reporting 57, 58  
Reset and Restart 17  
Resizing Time Graph 23  
Round Trip Time 118

## - S -

Save on exit 118  
Scrolling 20  
Service 52  
Show on summary graph 118  
SMTP Password 120  
SMTP Server 120  
SMTP Username 120  
Sort 23  
Summary Graphs 23  
Support 133

## - T -

Time Graph 18  
Timeline Graph 18  
Timeline Graphing 20  
Timeout speed 125  
Tool Tray 118  
Trace Address 18  
Trace Graph 18  
Trace Interval 17  
Trace Route 12  
TTL 12

## - U -

Unix Agent 105

## - V -

VoIP Troubleshooting 89

## - W -

Web interface 27, 119  
Windows Agent 105  
Windows Service 52  
Workspaces 25

