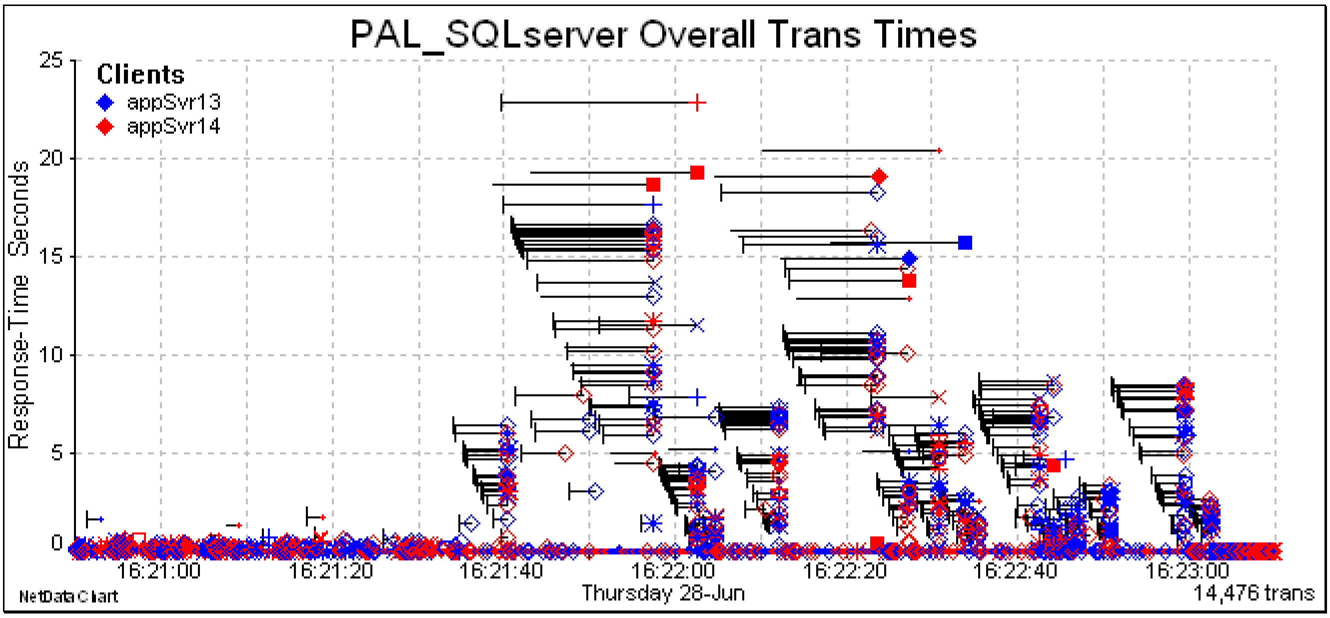
Measure IT Pty Ltd

NetData

visualising IT performance



Charting Guide

Getting Started with NetData Lite

Commercial-in-Confidence

This document contains commercially-sensitive and proprietary technical information. Measure IT requires that it not be released in any form to any personnel other than NetData licensees and those addressed explicitly and directly by Measure IT.

Contact Measure IT: Bob@netdata-pro.com

CONTENTS

Getting Started with NetData Lite

NetData is easy to install and requires no changes to the Windows Registry. It can be run from a USB key.

NetData is supplied as a zipped executable and a zipped set of run-time files. Unzip both zip files into a NetData folder.

We suggest that you place an icon on the desktop to launch NetData.

The simplest way to analyse a capture file is to drop it onto NetData. NetData will then automatically analyse the capture, build its database and display a variety of charts that depends on the size and content of the capture file. Charts can be closed and re-opened at any time with the large buttons on the main window, or from the View menu.

What do we do when the charts appear? Three of the charts show how the system behaves over time, a little like a slow-motion video. We can fly over the broad landscape, noting any unusual patterns, or zoom into an interesting feature and step through the details of its transactions and packets. We have two objectives: to better understand how the system and its applications operate; and to identify any form of unhealthy behaviour. There may be evidence of a blockage, inappropriate settings of TCP flow-control parameters, or consistent circumstances of packet loss; there may be signs of stress that indicate a heavily loaded server resource. We can identify and characterise small problems before they become serious, as well as the already serious problems that require immediate remediation.

The copious arrays of chart overlays and controls may be forbidding, but they simply reflect NetData's wide range of analytical techniques. Some may be needed only rarely, but each one can play a crucial role in the diagnosis of some performance issues. Above all, NetData has evolved to make it as easy as possible to move between charts and tables that give different views of a problem, to corroborate evidence and test hypotheses. The following notes describe the more frequently used charting functions and some of the short cuts for exploring chart features and navigating between charts.

There is a huge wealth of diagnostic information in capture files, and through these charts NetData presents particularly detailed and accurate pictures of system operation that can be drawn *only* from network traffic.

"Without packets and NetData, you're just floundering in the dark"

Charting Overview

NetData has four main charts that can be varied and customised with a large array of controls reached through a button bar just above each chart. All but the dialogue chart have their own window of formatting controls accessed with the Format button which is the first on the bar.

When examining a capture for the first time, charts are usually examined in this order:

Dialogue Chart: summarises all the dialogues in the traffic, identifying the clients, the servers, and the protocols handled by the services accessed through each active port on each server. Dialogue line thickness and colour indicate traffic volumes and rates of various network abnormalities.

Performance Chart. We first look at the volumes of different types of traffic and note any red bars of *Missed* traffic that indicate the volume of traffic dropped by the sniffer. A large percentage of missed traffic introduces a level of uncertainty in our interpretation of all the charts.

After examining traffic volumes, we display the chart’s format-control window (with the Format button) and uncheck ‘Traffic volumes’ to remove them from the chart. We can then focus on other performance information.

This chart’s main function is to display the response times of individual transactions, and we often start with the largest response times, examining their circumstances – perhaps by displaying the transactions on the timing and data-flow charts – and then hiding them from the performance chart. As we successively hide the transactions of particular types or servers – by right-clicking on markers – the chart’s scale changes automatically to make other transactions more prominent.

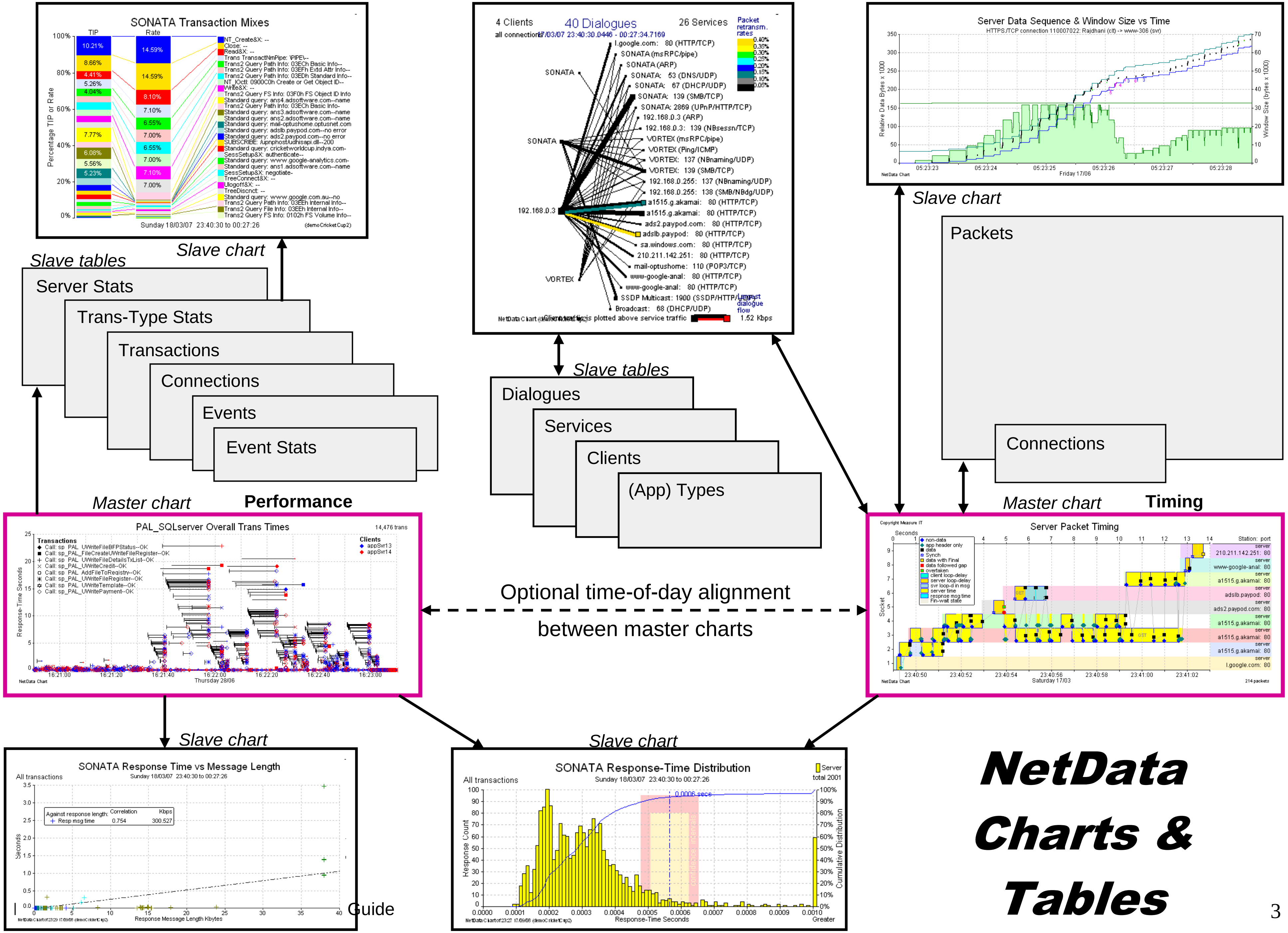
Timing Chart: displays the transaction activity of any set of connections. Each transaction is displayed with horizontally-stacked bars, breaking transaction time into a variety of components that identify request-message transfer, service, response-message transfer, propagation delays and retransmission timeouts.

The timing chart can be overlaid with markers for the packets in every connection, and they reveal how network abnormalities contribute to transaction delays. Each *connection band* is now split into a pair of *socket bands* with the same pastel colour; the lower band carries only the markers of client packets, and the upper band carries only server packets. The height of a packet’s marker within its band is proportional to its length. Normally, a grey line joins markers in chronological order, and other lines can join data packets with their ack packets, and retransmissions with their original packets.

The transactions on a timing chart can be rendered alternatively in any one of four types of waterfall chart.

Data-Flow Chart: has two major modes which draw graphs from the packets of either a single connection on the timing chart, or all connections on the timing chart. The first mode focuses on the flow-control behaviour of a single connection and, with its highly detailed graphs of sliding windows and various views of window size, can reveal virtually any cause of low throughput.

The second mode is designed to characterise the complete flow of a network channel and investigate causes of packet loss such as microbursts (queue overflow), packet shaping and packet policing.



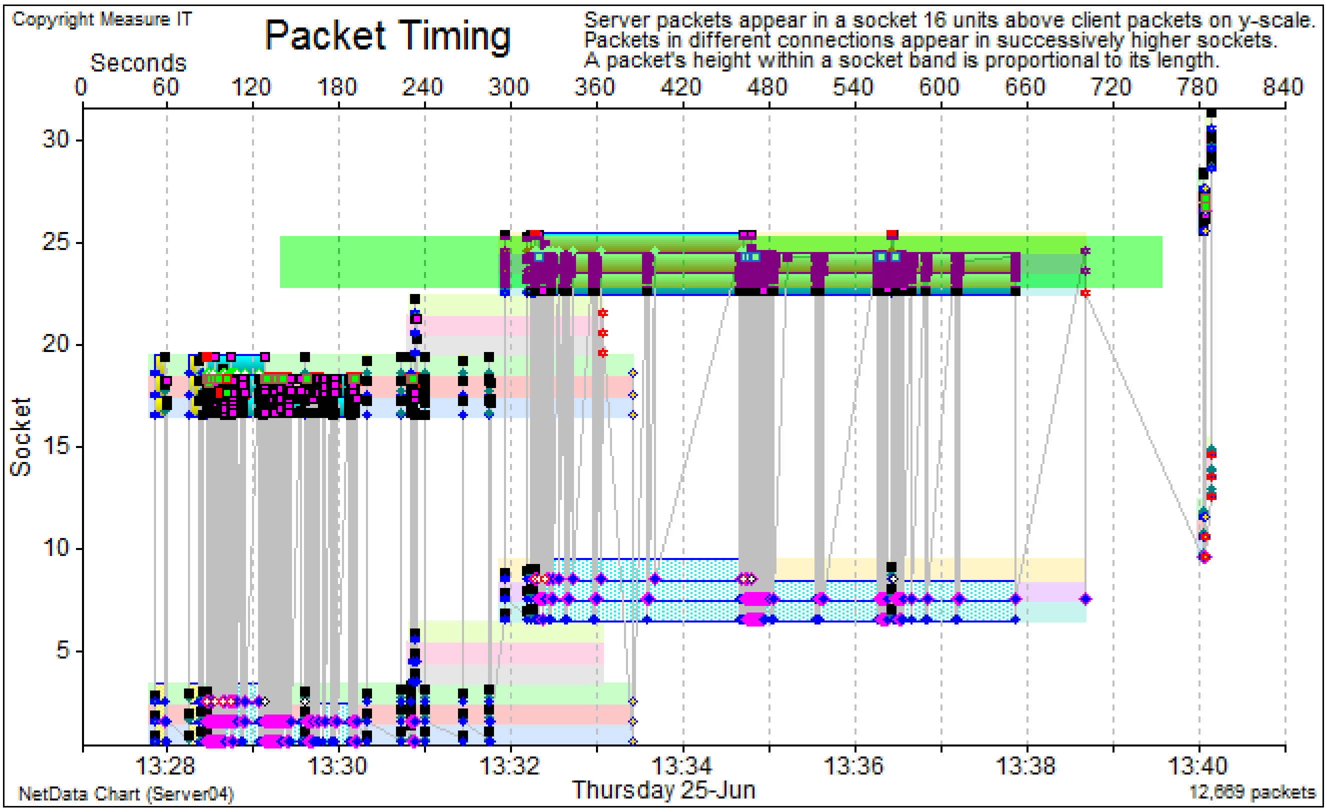
Guide

NetData Charts & Tables

Zooming

There are many ways to zoom into and out from any part of a chart. The quickest is to drag the cursor over the desired region of the chart, creating a green rectangle to identify the region. Initially the rectangle is painted red, indicating that the zoom is not yet armed, to avoid accidental zooms. The rectangle changes to green when it reaches a minimum width, and stays green even if the size is reduced.

Normally, the dragged rectangle spans the full height of the chart and determines only a new time span. The **Sel. Conns** button on the timing chart, and the **Fixed Seq. Scale** button on the flow chart, toggle the zooming mode to change the vertical scale as well as the horizontal time scale.



The green zoom rectangle on this chart restricted the time to range from 13:31 to 13:39 (with auto zoom and snap-to-grid) and also confined the chart to three of 15 connections.

The quickest way to **zoom out** is to click on the **Back** button to undo a zoom. The **Fwd** button will redo the zoom, and together the **Back** and **Fwd** buttons act like the corresponding arrows on a browser. These buttons are reset and lose their memory of recent changes when new records are loaded from the database.

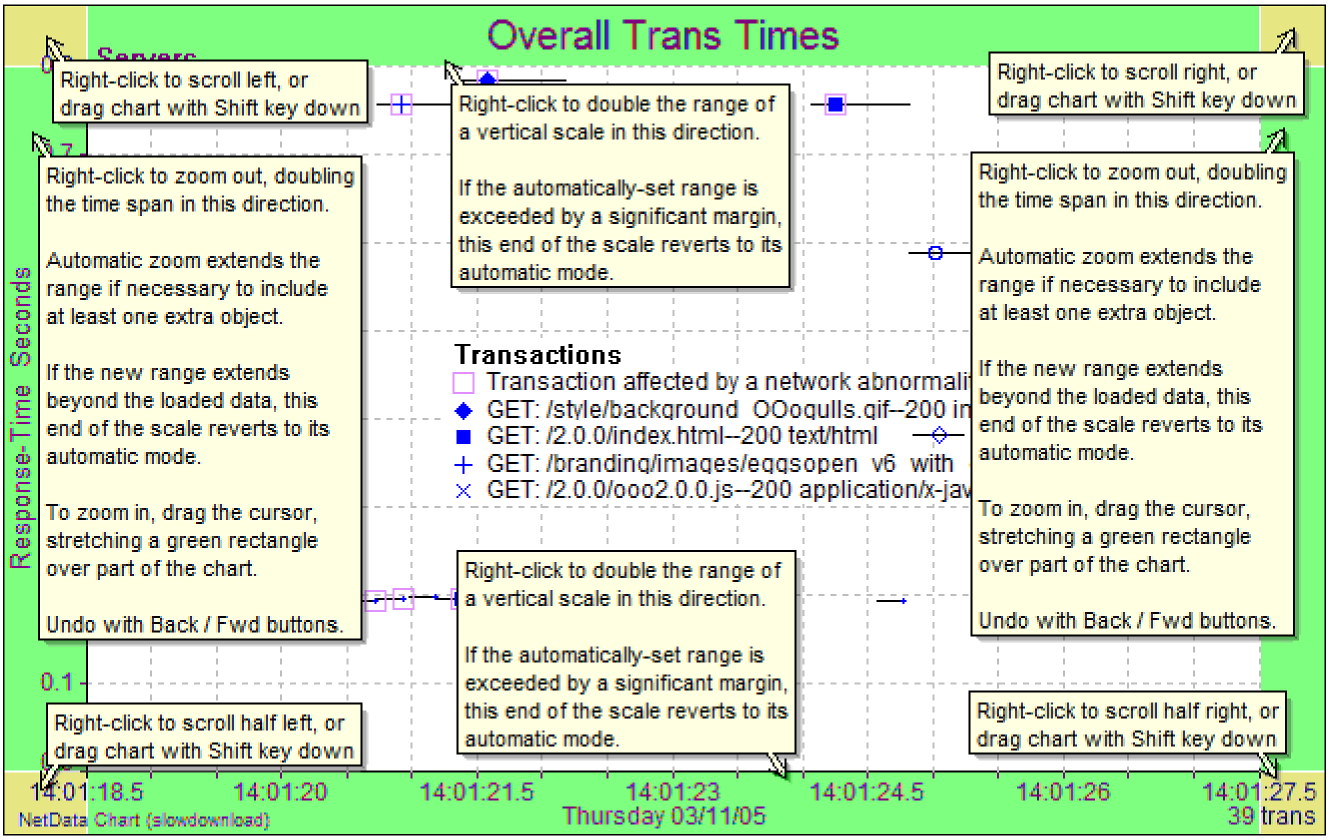
Another way to zoom in is to right-click on the chart and, from the context menu, choose to make the selected time either a new start-time or a new end-time. To zoom out, the chart's time span is doubled, expanding to the left or the right, by right-clicking outside the grid area.

Changes to chart scales are normally subject to snap-to-grid and auto-zoom policies. The latter policy broadens a zoom-out function until at least one new object appears on the chart, and it extends a zoom-in function to eliminate empty grid panels at the end of the chart. These policies can be disabled in the chart's format-control window.

Scrolling

Horizontal

After zooming into a short time span it is often desired to scroll the chart forward or back, to study successive time periods. This is best achieved by right-clicking in a corner of the chart, outside the grid area. The top corners scroll the chart by its full time span, and the lower corners scroll the chart by half the time span.

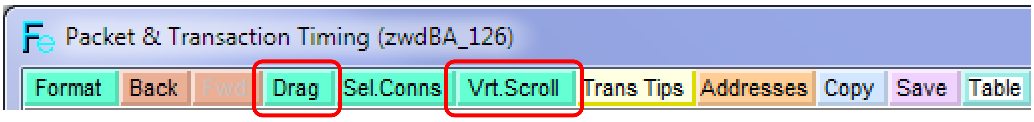


Right-clicking any of the eight distinct regions outside the chart's grid causes the chart to scroll or zoom out.

A fine adjustment can be made by dragging the chart with the left mouse button and the Shift key down.

Vertical

If the timing chart has a large number of connection bands, or a waterfall chart has many rows, their details can be made more readable by clicking the **Vrt. Scroll** button. Vertical scrolling is assisted by a scroll bar on the edge of the chart, and the chart can also be dragged with the left mouse button if the Shift key is down.

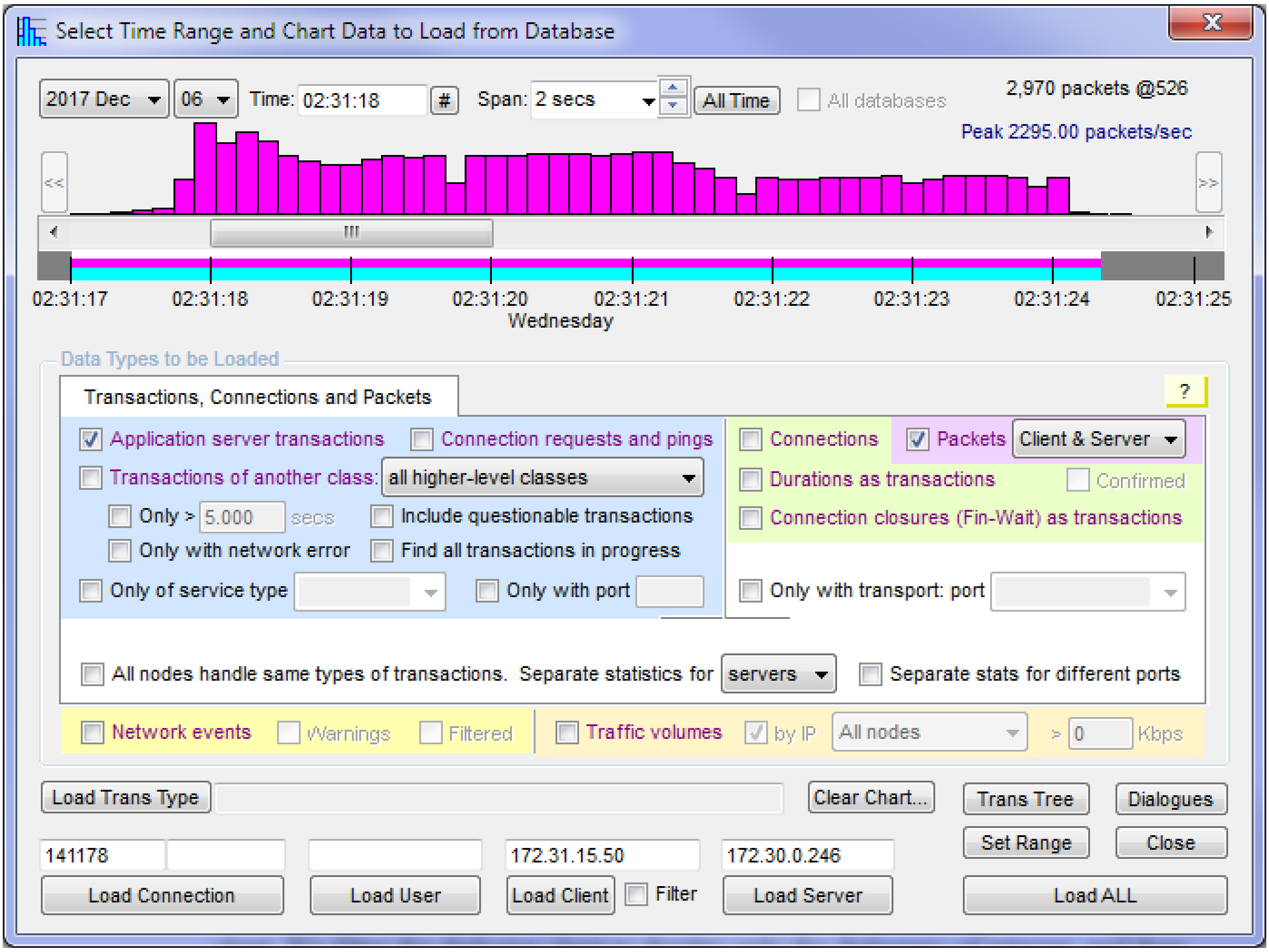


The **Drag** button toggles the cursor-drag function between zooming and scrolling actions. In the drag-scrolling mode the chart can be dragged without using the Shift key.

Data Loading

When NetData displays a chart immediately after analysis it loads from the database all the conventional types of records across the full time span of the captured traffic. Extra records can be loaded at any time to add overlays to a chart. For example, to overlay the performance chart with graphs of the numbers of concurrent connections held by each server, simply load all the connection records in the database. When new records are loaded, NetData always asks whether the existing records in the charting module are to be retained or cleared, to generate completely new charts.

The standard way to specify what records are to be loaded is with the **load-data window** which is displayed by the **Load** buttons above the performance and timing charts. A scroll bar at the top of the window controls the time range of loaded records. Click the **All Time** button to span the whole database, or click an adjacent spin button to reduce or increase the time span. The hash button (#) aligns the scroll bar with a notional chart grid.



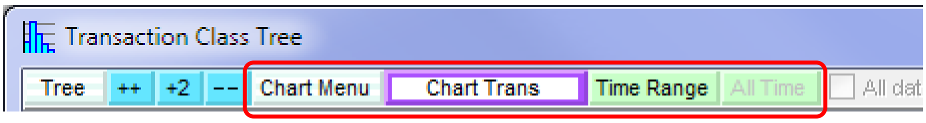
Check-boxes with purple legends in the middle portion of the load-data window specify the types of records to be loaded: server transactions (single round-trips), connections, packets, network events or traffic volumes, for example. Other controls specify filters that might refer to a port number, application type, or minimum response time. Most controls have a pop-up that describes its function.

Loading is initiated by one of the wide buttons at the bottom of the window. Five of these buttons specify an extra filter, constraining records to a particular transaction type, connection, user, client or server.

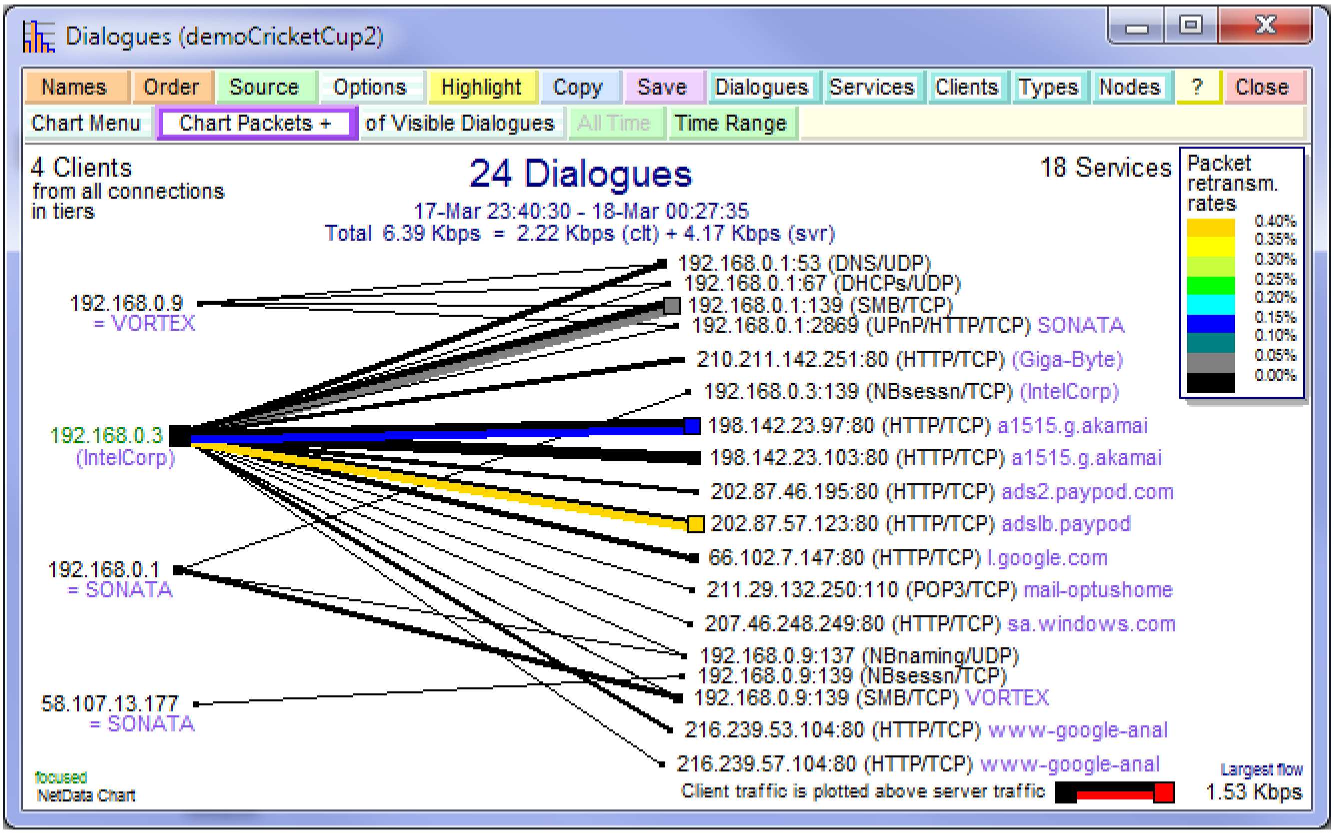
NetData's **object focusing mechanism** saves us from typing the names of filter objects. Each time we ask for a description of a transaction or packet, NetData notes our interest and sets a focus on that object. Alternatively, we can right click any object on a chart, or any row in a table, and ask for the focus to be set on the selected object. Focused objects appear above their respective Load buttons at the bottom of the load-data window, allowing us to load only the records of a focused object.

Loading the records related to a *group* of clients or servers is easier from the **dialogue chart**. The context menus of that chart have extensive options for hiding or revealing selected clients, servers, services and application types on the dialogue chart. We filter the dialogue chart to display only the dialogues of interest, and then click the Chart button on its button bar to load only the records related to the visible dialogues, the visible clients, or the visible services. The types of loaded records can be set with the menu under the **Chart Menu** button. The loaded records populate the performance and timing charts.

Record loading can also be initiated from the **transaction-class tree** that can be selected from the main window's View menu. The tree can be searched to find all the transaction types with a common element in their descriptions – such as web requests returning a significant HTTP status code, or SQL queries accessing a particular table – and the tree's Chart button will load all the records of only the relevant transaction types. The transaction tree and the dialogue chart work in much the same way, with similar buttons for the chart menu, setting the time range, and initiating the loading of records.



Dialogue Chart



Colours on the dialogue chart highlight those dialogues with the largest rates of a selected type of network abnormality. A ‘rainbow’ box of legends, normally parked in the chart’s top-right corner, indicates the range of error rates associated with each colour. The type of abnormality is selected from the menu under the **Highlight** button.

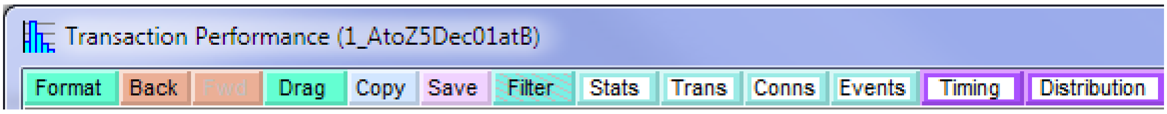
The menu under the **Options** button applies broad filters to the chart. Two options transform the dialogue chart into a bar chart that displays node connection-setup times, or displays top talkers according to a variety of attributes such as volume (Kbps), packet rate, and error rate.

There are three types of dialogue chart, selected by the **Source** button. The first is drawn from all the connection records in the database and splits each server into its respective services (i.e. ports).

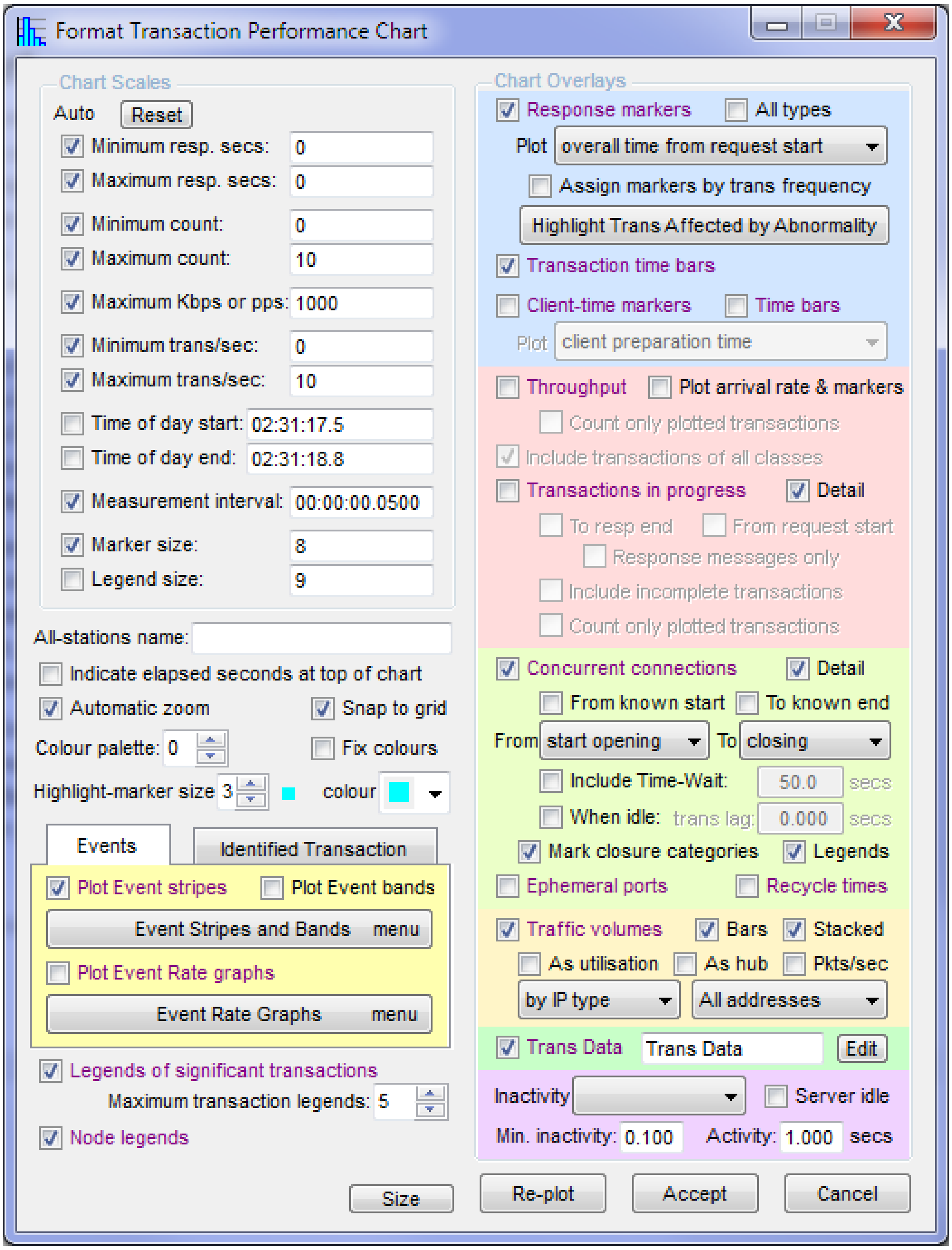
The second dialogue chart is drawn from dialogue summary records that aggregate the traffic across all the ports of a server. This chart highlights various aspects of the traffic capturing process such as the dialogues whose packets have been recorded twice.

The third dialogue chart is drawn only from the traffic displayed on the timing chart and is displayed by the **Dialogues** button above the timing chart. It is normally just a slave to the timing chart, but interaction between the charts is two-way. When clients, servers or particular application types are filtered out of this dialogue chart it changes the content of the timing chart to match.

Performance Chart



This chart can be overlaid with any combination of markers for response times and client-preparation or -reaction times; graphs of transaction rates (i.e. throughput), transactions in progress (TIP; i.e. transaction queue length), and concurrent connections. It can also display stacked bars indicating traffic volume by protocol type or IP address. The occurrence of network events, if loaded, can be indicated by vertical stripes of different colours. If an event rate is high it can be plotted as a line graph.



The format-control window for this chart has a standard layout. The group of controls at the top-left relate to chart scales. When a scale is in automatic mode it covers all the data loaded from the database, and the associated value is the extreme data value that determines the scale. There is rarely any need to enter a scale value because all scales can be adjusted by clicking at appropriate points on the chart.

The controls on the right enable and customise overlays. The controls for the six different types of overlay are grouped together by the different background colours. Each overlay is enabled by a checkbox with a purple legend, and most overlays are enabled by default, to ensure that the overlay appears as soon as the relevant records are loaded from the database.

The **Accept** and **Re-plot** buttons on the bottom of the window accept new settings, but only the **Re-plot** button re-plots the chart immediately. The **Size** button presents a menu for setting the chart to a standard size suitable for reproduction at original size, on a page in either portrait or landscape orientation.

The performance chart normally displays what NetData calls *overall* response times, times that are measured from the start of a transaction's request message to the end of its response message. This time is indicated by the transaction marker's height against the response-time scale on the left, and, depending on the scale, it may also be indicated by a horizontal line running through the marker, measured against the bottom time-of-day scale. The marker is plotted at the time of day at which the response message started. If the request message is conveyed in more than one packet, a vertical tick is plotted on the horizontal line at the time of day at which the request ended. The horizontal line is then divided into the three main components of the overall response time.

A drop-down menu at the top of the format-control window redefines the plotted response-time measure, A common alternative to *overall* times is to plot just the server processing times.

Client-time markers are often plotted when investigating backend problems involving large bursts of round-trips. The normal response-time markers indicate processing delays in the backend server, and the client-time markers indicate abnormal delays in, say, a client application server when it digests a database response and prepares the next database query.

Overlaid graphs of throughput (transaction rate), transactions in progress (queue length), concurrent connections and traffic volumes are largely self-explanatory, but graphs of transaction data and server inactivity warrant an introduction.

NetData's application decoders extract from transaction messages some significant data fields such as database query targets, and this data appears as lists of text strings and parameters in the Data column of the transaction table. The charting module can select particular numeric parameters from the data of specified transactions and plot their values on the performance chart.

A subset of transactions is identified by selecting a common field in the transaction table and choosing to 'Plot Data of Similar Trans'. A subsequent dialogue window allows us to specify which item in the data list is to be plotted. This function is often used to plot the progress of a file pointer when a file is read or written with SMB Read or Write transactions. We identify the relevant transactions by selecting the description field of one transaction, and, if we choose a graph type of 'Levels', NetData automatically extracts the file pointer. It is also possible to plot markers indicating the lengths of the SMB data blocks, and a graph of the data-reading or writing rate.

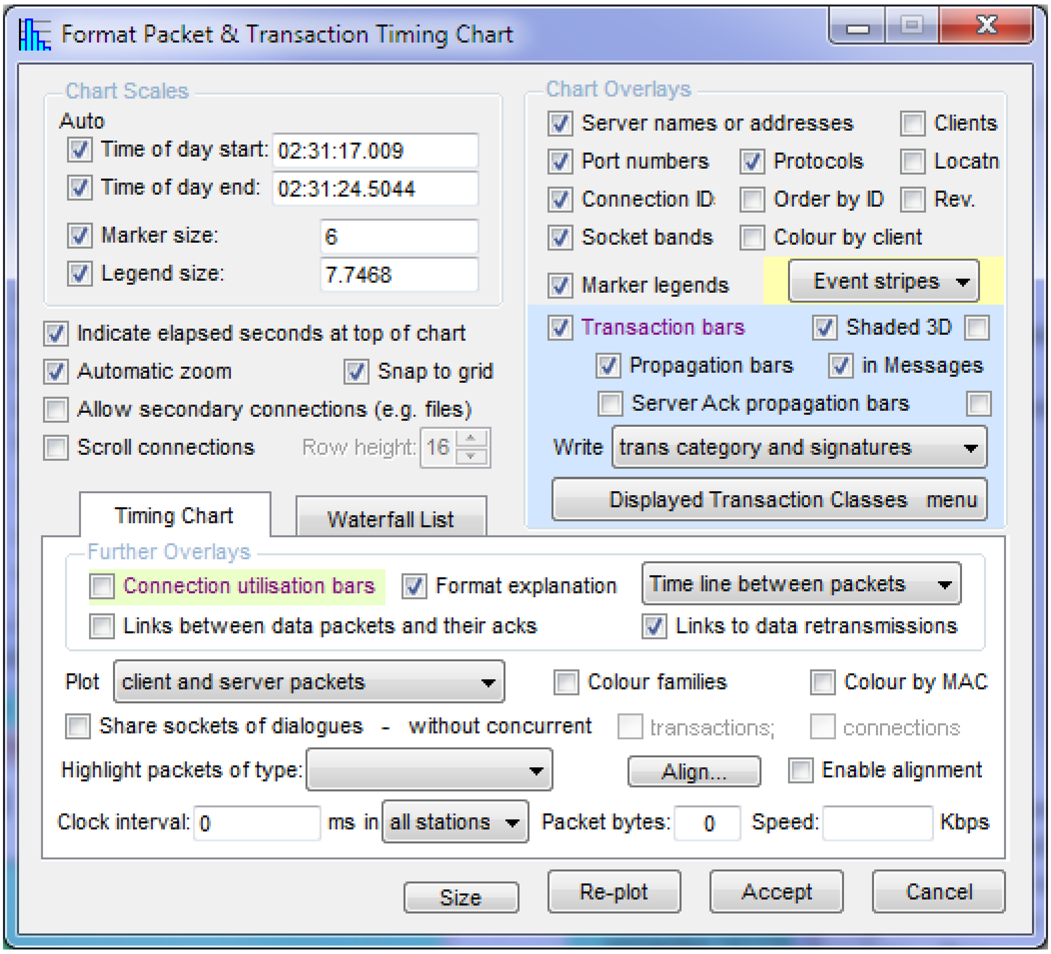
The **Inactivity overlay** helps to quantify the effect of garbage collection in, say, a Java Virtual Machine (JVM). It identifies periods of server inactivity that meet specified criteria, and displays the percentage of idle time averaged over the chart's time span.

Timing Chart

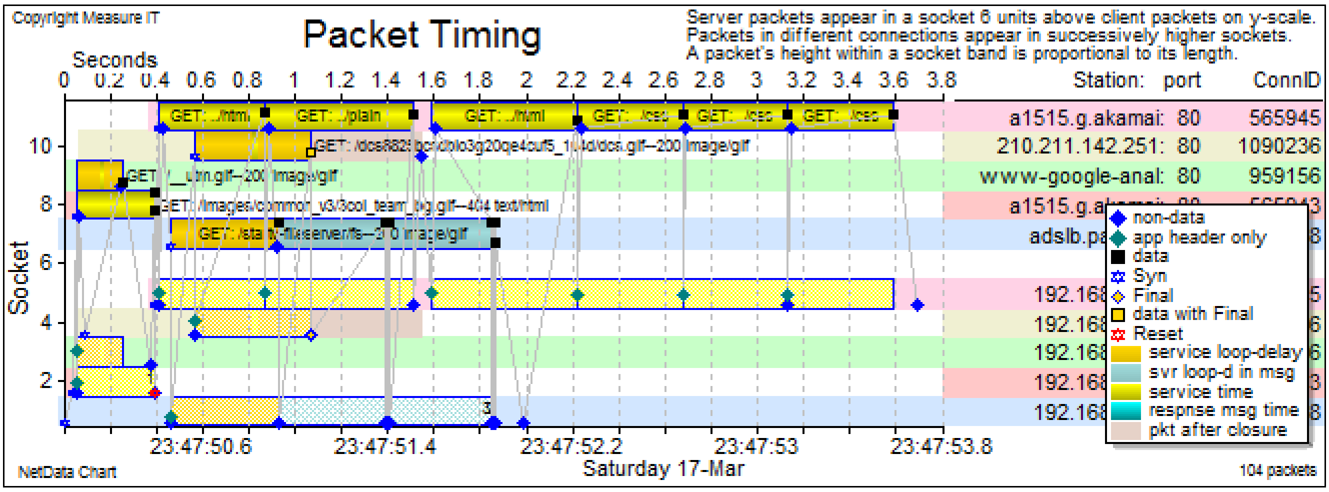
If the timing chart displays a single transaction, with packet markers on the transaction bars, it clearly shows how each packet progresses the transaction. If a packet was lost, it shows how the transaction was delayed while the data was recovered.

If the chart shows all the connections between an application server and a database server, together with the app server's front-end transactions, then it depicts the activity of virtually every application thread in the app server. A rapid burst of backend transactions in a single connection usually indicates that a thread is handling a front-end user transaction. The context menu of the timing chart provides an arsenal of functions to compare backend bursts with front-end round-trips and relate them as a transaction family.

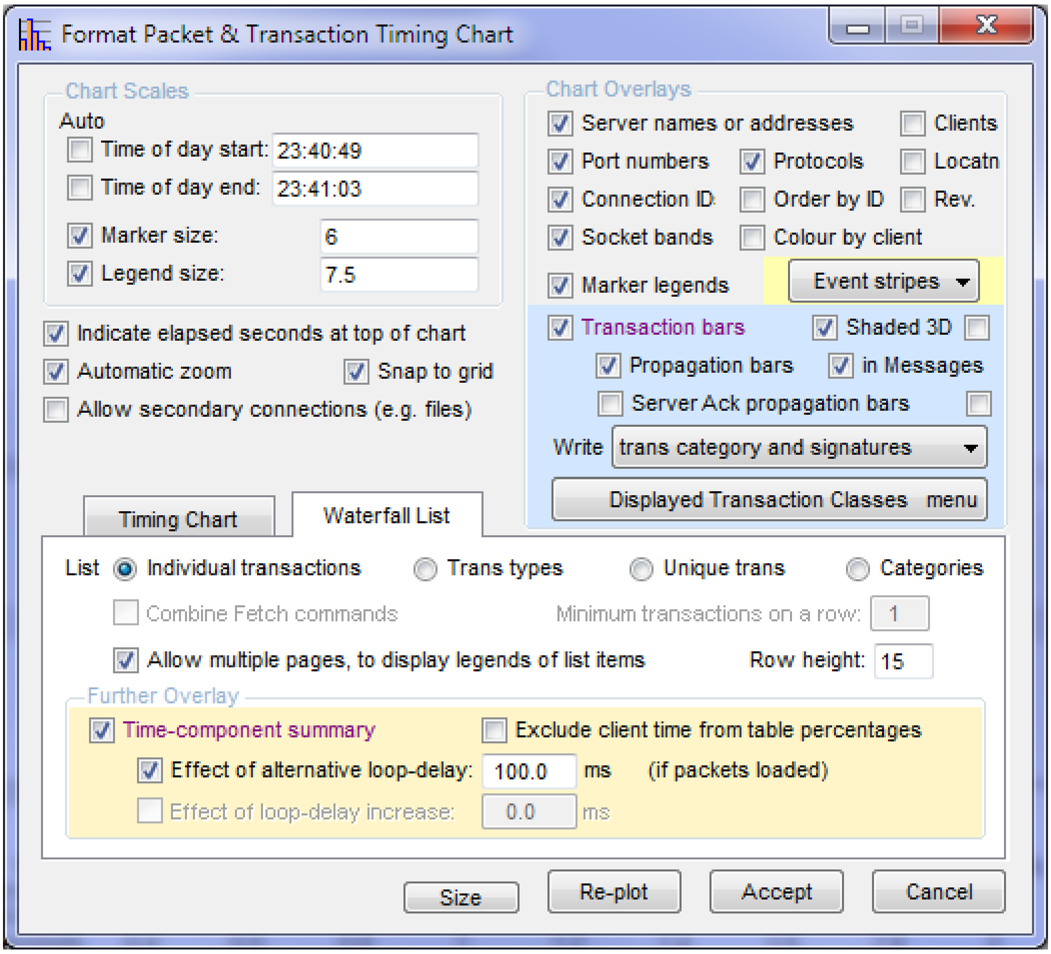
The first group of overlay controls in the chart's format-control window control the connection descriptions that appear on the right of the chart, and the pale-blue group controls the appearance of transaction bars.



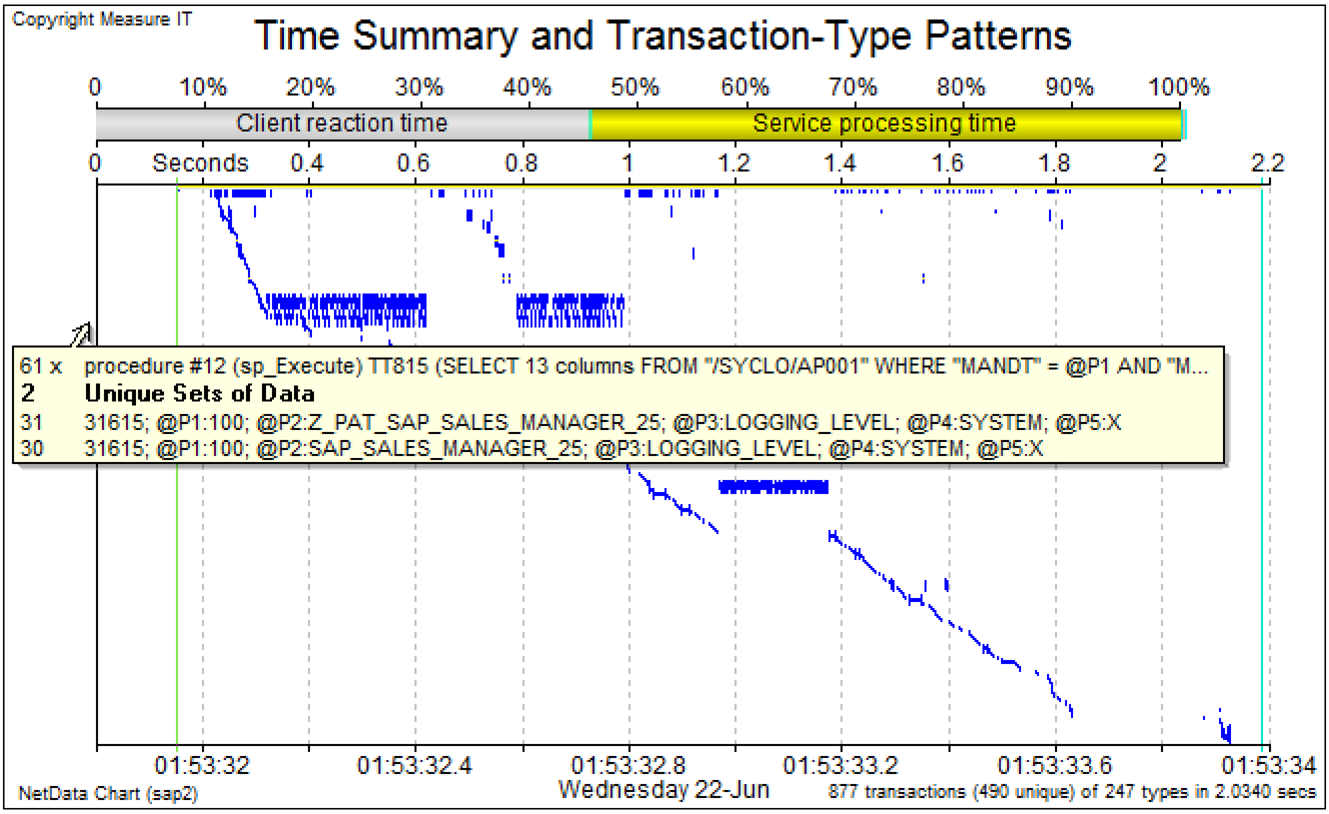
Two tabbed pages of controls in the chart's format-control window allow the transactions on a timing chart to be rendered in the form of a waterfall.



Waterfall Chart

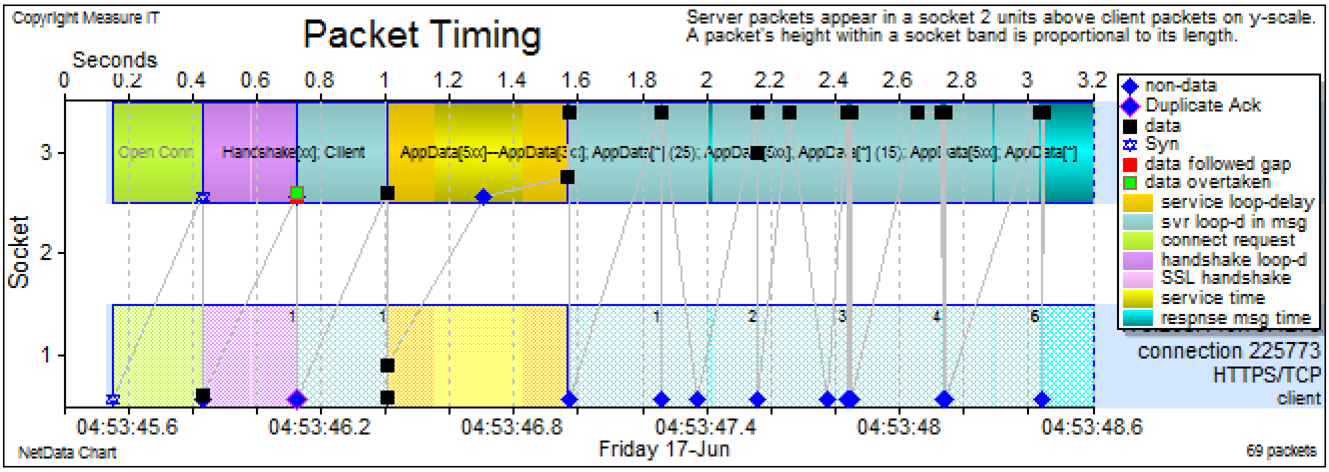


The traditional waterfall chart displays a single transaction on each row, but when there are many transactions, such as a long burst of database transactions needed to complete a user transaction, we gain more insight into the application thread's program by allowing many transactions on each row but restricting each row to transactions of the same type. The resulting pattern of transaction bars often identifies program loops and the types of query executed in each loop iteration. NetData can display the search parameters of each query and will highlight the execution of identical queries and other forms of inefficiency.



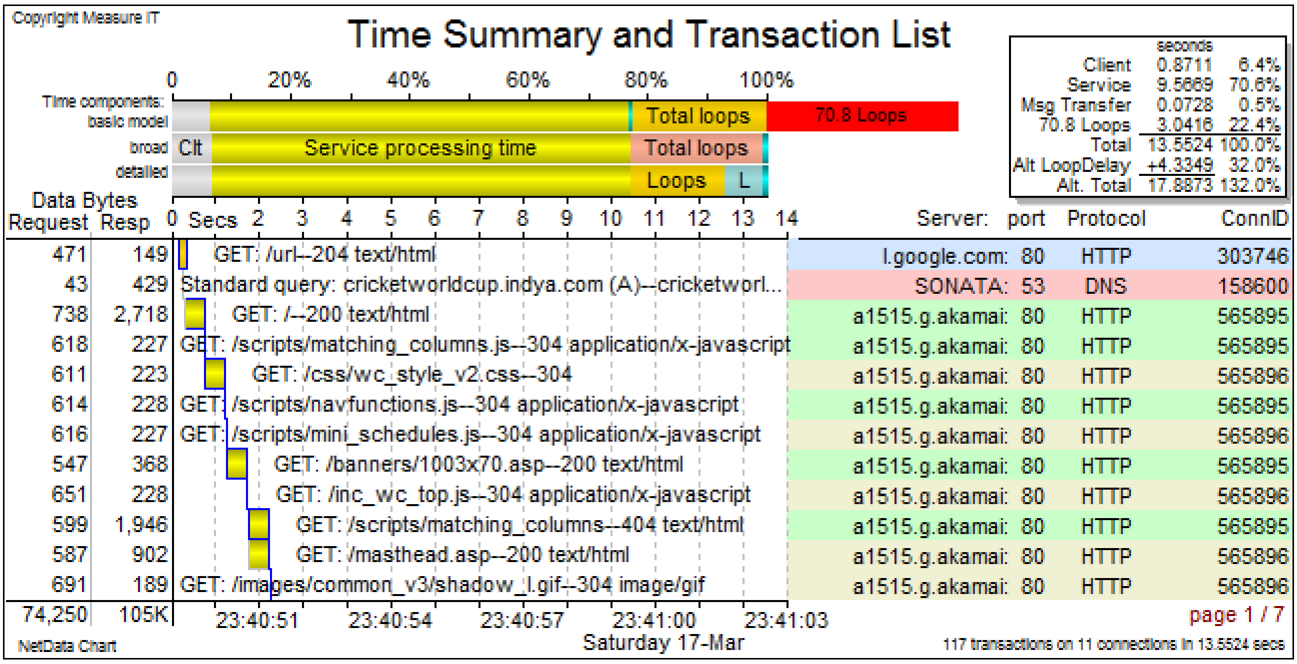
The pop-up on this waterfall chart refers to a row of 61 calls to the same type of database query, with only two unique sets of search parameters. In other words, 59 queries were redundant.

If packets have been loaded, NetData is able to count all the non-overlapping round-trip propagation delays, while TCP waits for acknowledgements and for request-response pairs. It can then calculate the impact of a change in the round-trip time on the overall response time, when the client or server is relocated.



This timing chart of a single transaction with a large response message shows how NetData counted five non-overlapping TCP round-trips during the message transfer. The propagation delays in these round-trips are plotted as grey-blue bars. Green, purple and orange bars identify propagation delays in different circumstances. On a waterfall chart this transaction, preceded by connection setup and SSL handshake, would contribute a total of 9 loops.

Stacked bars above the waterfall chart summarise how time is spent.



The red bar with the bar chart and table above this first page of a waterfall chart indicates the effect of a change in the network's loop-delay on the user transaction's overall response time. The critical statistic is the number of loops (round-trips) presented in the summary table. Because the user transaction has kept many connections active, NetData first identifies a single thread or path (like a *critical* path) of non-overlapping activities and then counts the various types of round-trips along that path. The bars of transactions on the critical path are displayed with blue borders.

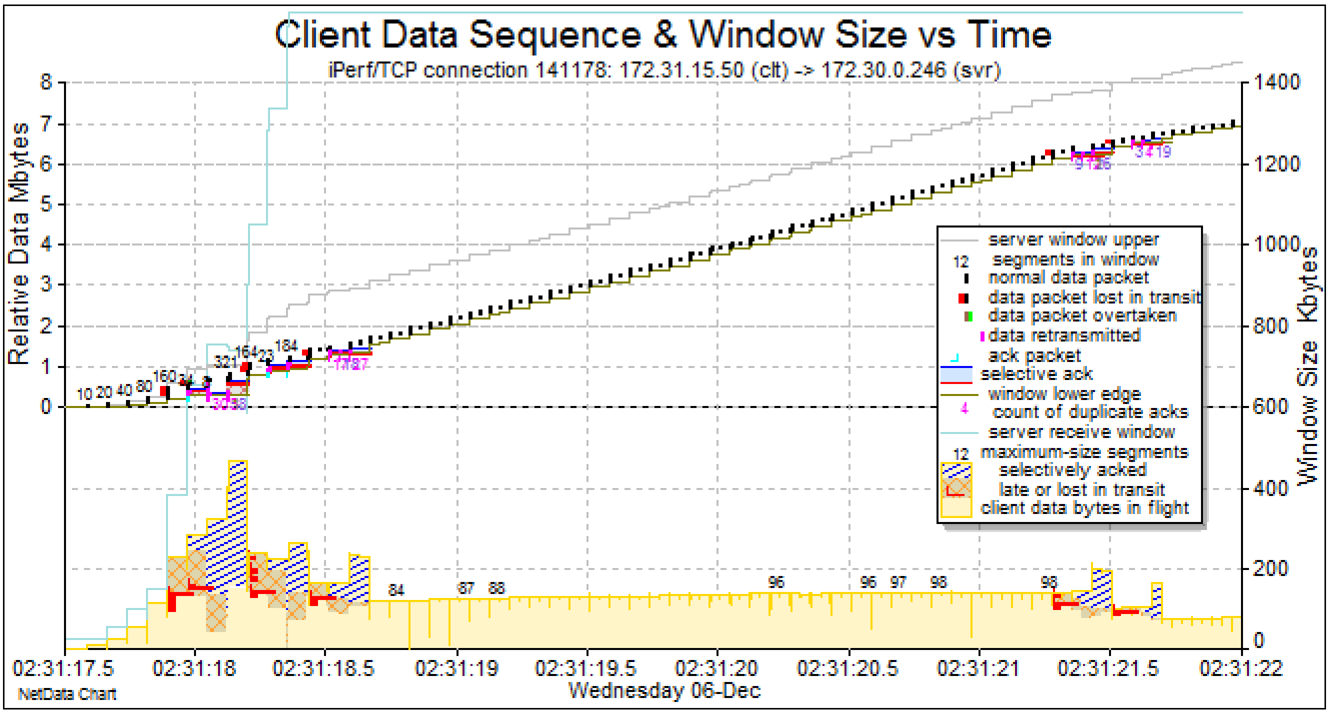
Data-Flow Chart

The data-flow chart is toggled between its single-connection mode and the all-connections mode by the **All Connections / One Connection** button.

Single-Connection Mode

Normally, the connection's TCP sliding window runs diagonally across the chart, and can overlap the sender's congestion window – more precisely, its data 'in flight' (waiting for acknowledgement) – which sits on the bottom of the chart.

There are two ways to separate and clarify the two parts of the chart: click at an appropriate height on the chart and choose to set the position of either the sliding window's lowest data sequence, or the bottom of the sliding-window graph. The latter option is preferred because the sliding-window scales remain in automatic mode and the chart adjusts sensibly when its time span is changed.



The bottom of the sliding-window graph has been aligned with a grid line of the window-size graph, at 600 Kbytes.

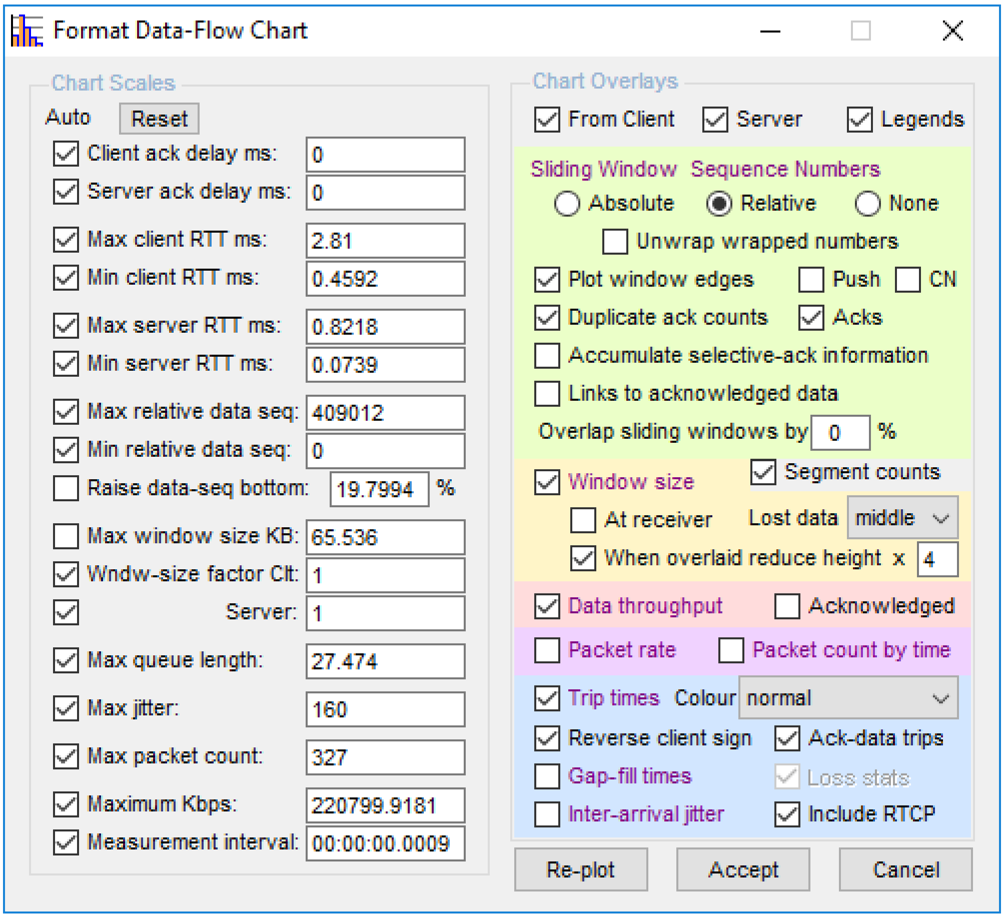
Initially, the data-flow chart displays the behaviour of both the client and server packet streams and their graphs may overlap. In most situations there is interest in only one stream and the chart is simplified by unchecking either 'From client' or 'Server'

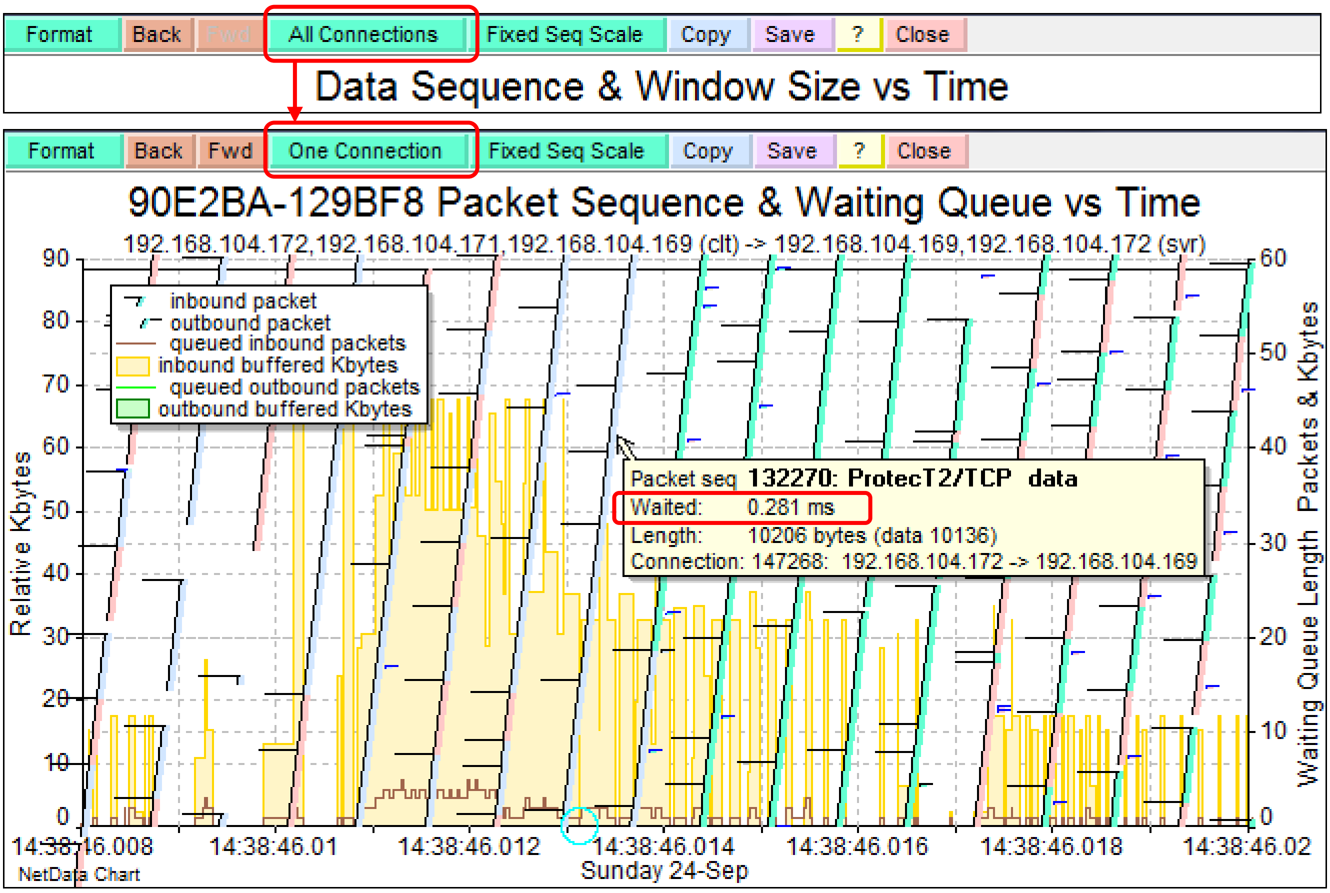
The sliding window graph includes vertical strips representing each packet, and stepped lines indicating the two edges of the window. The lower edge indicates the sequence numbers that have been acknowledged, and it may be augmented with pale blue and brown bands that represent the contents of successive selective acks. Different colours distinguish between SACKs and D-SACKs. It may be useful to check the box 'Accumulate selective-ack information', to carry forward the ack information in early SACKs that TCP can't repeat in later SACKs.

The window-size graph also reflects the contents of selective acks, indicating how much of the data in flight has been lost and how much has been selectively acknowledged.

The most valuable aspect of the window-size graph is that it estimates window occupancy from the sender's point of view. If traffic is captured on a node that is receiving a flow of data, then the packet timestamps will show that received data is acknowledged almost immediately, and there is rarely any significant data 'in flight'. NetData delays acknowledgement information by the loop-delay – the minimum round-trip time – from the sniffer to the sending node, to calculate the more-informative view of send-window size. This allows us to understand the flow-control rules governing the sender and recognise the many causes of slow transfers such as small window size and limited send-buffer space. NetData's measurement of a connection's loop-delay appear at the top of the chart's scale controls, where the amount by which acks are delayed can be overridden manually.

If the timing chart displays more than one connection, the data-flow chart may display a uniquely identified connection such as the connection in focus, but the displayed connection can always be set explicitly by selecting it on the timing chart and, from the context menu, choosing 'Plot Flow Chart of This Connection'.





A *single-connection* chart has been swapped for an *all-connections* chart that displays strips representing all the packets addressed to a nominated MAC address. The strips indicate an unbroken sequence of packets between 14:38:46.010 and 14:38:46.017, a microburst formed from the packets of three connections (with different colour strips). The underlying cream graph indicates the length of the queue of waiting packets.

All-Connections Mode

The all-connections mode allows us to see the round-trip times and aggregate the bandwidth of all the packets displayed on the timing chart. NetData normally separates the client and server traffic flows, but if there are both clients and servers on each side of the network, these flows produce invalid measures of link utilisation. In such cases we aggregate all the traffic flowing in the same direction by nominating either an IP or MAC address as a ‘monitored’ address, at the top of the chart-overlay controls. Then, NetData separates the traffic into *inbound* and *outbound* streams.

When a sliding-window chart reveals that packets are lost near the end of long bursts, it suggests the action of a packet shaper or that a queue is overflowing its buffer space. A single connection can’t give a complete picture of packet bursts, but they can be viewed in the all-connections mode.

The simplest way to define the stream of packets heading for a queue is to right-click the marker of a representative packet on the timing chart, and in the Focus submenu choose ‘Destin MAC Adrs’. Then, in the flow chart’s format-control window, click the Focused button and choose the focused MAC address; it will be entered into the

Monitored Address box above the button. It is assumed that this is the address of the LAN gateway through which will pass all packets heading for the queue.

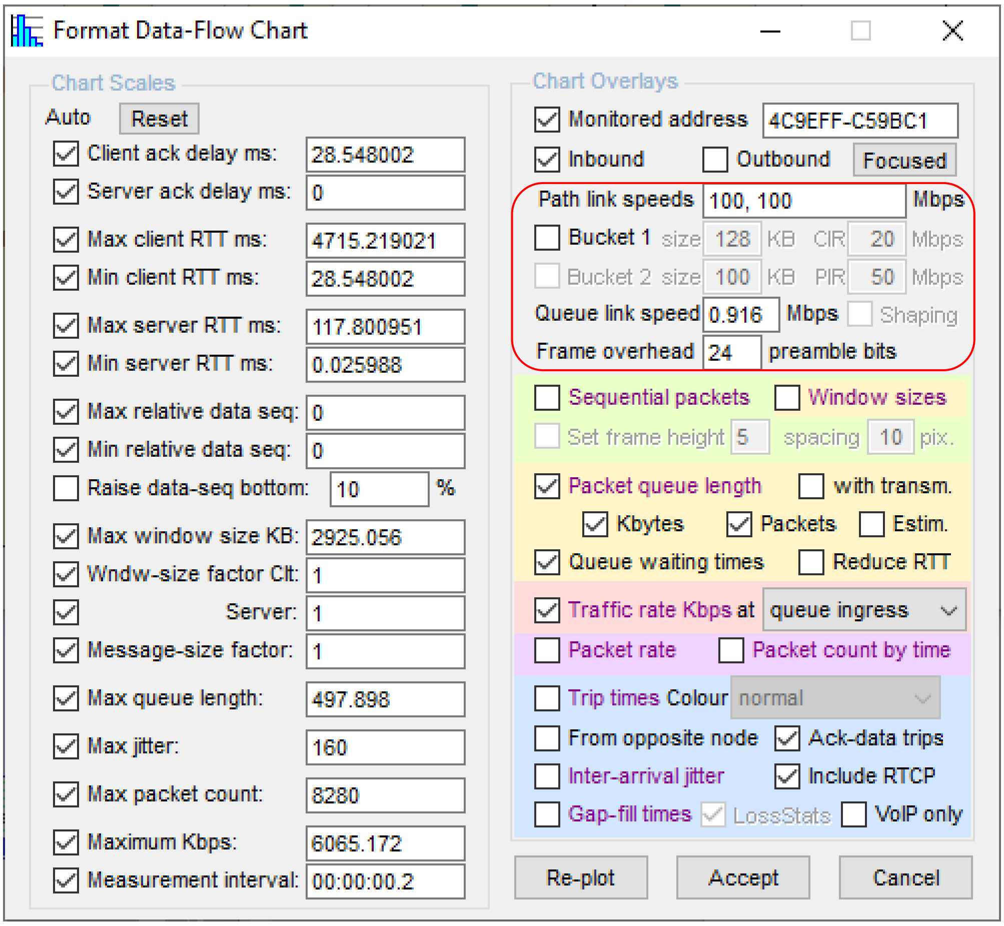
When ‘Plot all sequential packets’ is checked, the chart displays strips for all the packets of an inbound or outbound stream, along a segmented diagonal line with a scale that can indicate when a link is idle or busy. This chart depends on knowledge of the link’s speed, and it needs to be entered as the ‘Queue link speed’.

If a large burst contains many packets from one connection, the TCP flow-control rules generating the large burst can be investigated by swapping to the single-connection mode. Right-click on a packet in the complete burst and, from the context menu, choose ‘Plot Only This Packet’s Connection’.

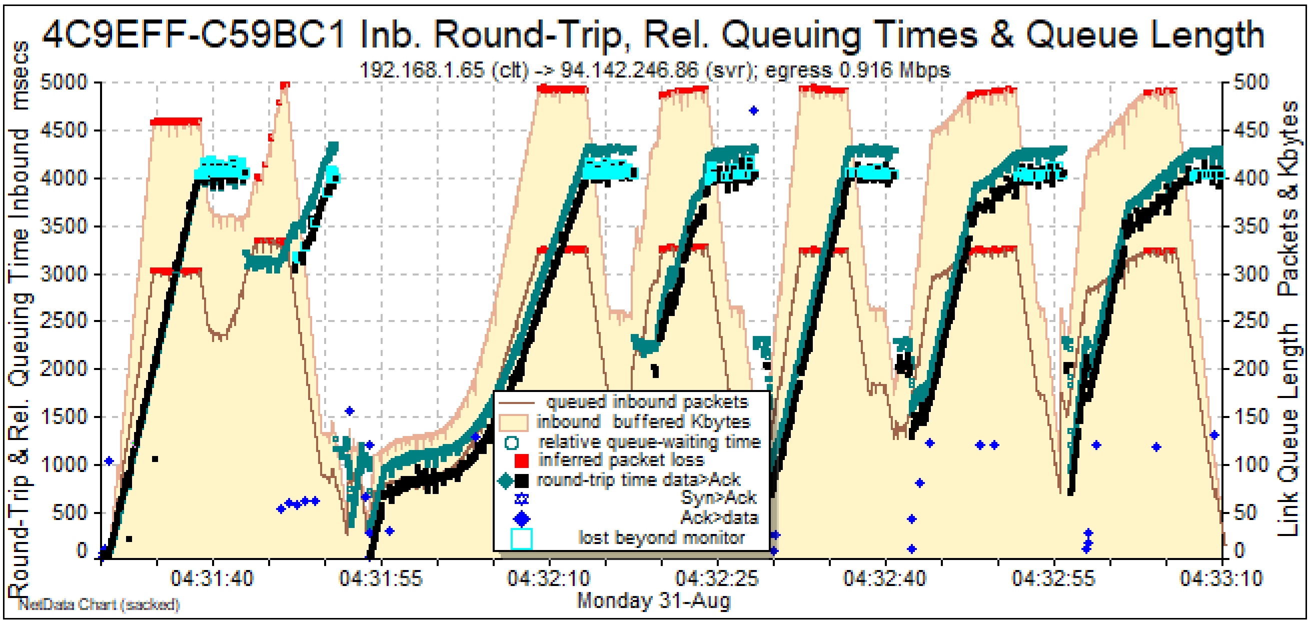
Queue Modelling

Besides displaying the contents of all the microbursts traversing the link, NetData also models the formation of queues as packets pass the sniffer and wait to be transmitted over the nominated link. The chart can overlay a graph of the queue length with markers plotted at the length of the queue when packets were lost. If the modelling parameters are correct, the chart will show packets being lost only when the queue reaches a peak length.

After passing the sniffer, packets may traverse many links before arriving at the queue that feeds the nominated link. If the speeds of all the intervening links are entered in the box for ‘Path link speeds’, NetData will model transmission of packets over the links, and the formation of queues between the links.



To determine the correct modelling parameters – primarily the ‘Queue link speed’ – the parameters should be adjusted until ideally the chart displays two confirming features as below: packets lost only at a uniform peak in the queue length (red markers); and within a packet burst a constant difference between the measured round-trip times (black markers) and the modelled, relative queue waiting times (green markers).



Most packets were lost when the number of packets in the queue (brown graph) reached the peak of 325 packets. Packets in the first burst were lost at an apparently shorter queue length but this anomaly could be explained by queued packets that were not seen by the sniffer. During that first burst the bands of black and green markers were aligned very closely, but the unseen packets produced a constant difference of about 300 ms between the marker bands in subsequent bursts.

Modelling Packet Shaping and Policing

Most routers, particularly at the edge of wide-area networks, provide extensive facilities for limiting bandwidth use and the size of packet bursts in specific conversations. The configuration of these regulation mechanisms is usually expressed in terms of delivery ‘contracts’, and in operation, after assessment by the regulator, each packet is assigned to one of three categories that are associated with different colours. A packet may be judged as conforming (green), exceeding (yellow), or violating (red) the contracted bandwidth. Conforming packets are passed for transmission without further delay, and violating packets are almost invariably discarded. Shaping mechanisms usually queue ‘exceeding’ packets for delayed release in the equivalent of a leaky bucket, whereas policing mechanisms may mark such packets with a different distributed-system code point (DSCP) or QoS which makes them more vulnerable to loss elsewhere in a congested network.

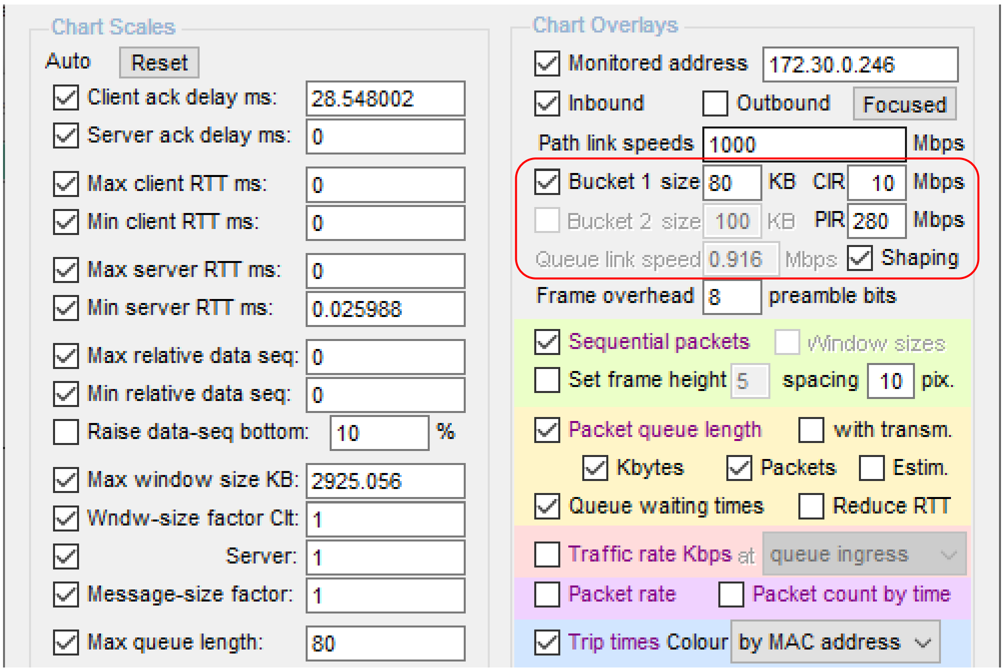
Regulators need to limit bandwidth averaged over long periods – the Committed Information Rate (CIR) – while allowing short-term peak rates (Peak Info Rate) and bursts of an acceptable size. Most regulators use some form of token bucket. Tokens are expressed in bits or bytes and are fed to buckets either continuously or at regular intervals at rates equal to the CIR or PIR. To progress well, a packet must be able to remove a token of matching length from a bucket.

The simplest mechanism implements a single rate with a single bucket. A second bucket may be added to hold tokens that overflow the first bucket, and, if a token is withdrawn from the second bucket because the first is empty, the packet is marked as 'exceeding'.

A three-colour policer usually involves two token buckets that are fed tokens independently at different rates, CIR and PIR. Conforming packets remove a token from both buckets.

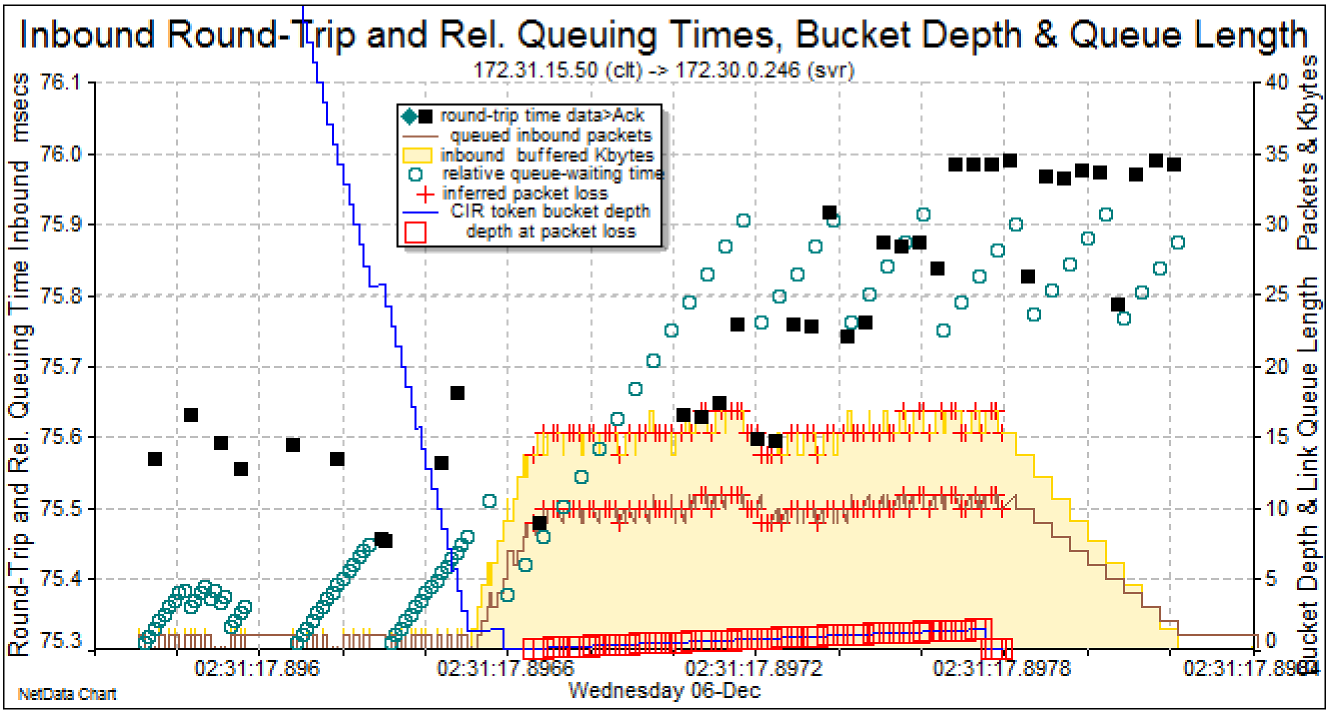
If the queueing model can't explain a packet loss, it may be caused by the action of a packet shaper or policer. To test such theories, NetData can model the depth of one or two token buckets, or the combined action of both a token bucket and a leaky bucket for packet shaping. Checkboxes enable the token-bucket models. When 'Bucket 1' is enabled, the Shaping checkbox enables the leaky-bucket model, to queue 'exceeded' packets for release at the PIR.

Modelling parameters include bucket size, a CIR for the first bucket, and a PIR for the second bucket or queue.



A second token bucket for a PIR is optional and can't be used if packet shaping is requested. The second bucket is also a token bucket. It is used to model traffic policing and regulates a PIR in what can be described as a Two-Rate Three-Colour Marker. A packet is forwarded only if a token can be removed from both buckets. The bucket's size is known as the Peak Burst Size (PBS). The model releases all packets to the optional queue irrespective of the token depth.

The pop-up tips for the two bucket checkboxes describe the modelling in more detail and summarise their packet-handling algorithms.



The modelled action of a router's packet-shaping policy is illustrated by the blue line indicating the depth of a token bucket. When that bucket became empty, arriving packets were directed to a queue in a 'leaky' bucket, and when its length reached 10, packets were dropped, as indicated by the red markers. The occurrence of packet losses at an apparent ceiling to a leaky bucket supports this theory for the packet loss.

The greenish circle markers plot the calculated queue-waiting times, and they rise by about half a millisecond as packets are directed to the queue in the leaky bucket. The change in calculated delay is consistent with the observed increase in network transit times indicated by the black-square markers, and gives further support to the packet-shaping theory.

Handy Tools

Pop-Up Tips

Almost every object and graph on a chart will pop up a description when the cursor rests near it. A pop-up can be pinned to the chart by typing Alt-Z. Pop-ups disappear when the chart is re-plotted for any reason, or Alt-R is typed. Typing Alt-X or Alt-Y toggles the preferred position for the current pop-up.

All pinned pop-ups are retained on the chart when it is saved to a disk file or copied to the clipboard for pasting into a document,

Investigating Transaction Details

When the performance chart attracts attention to a transaction with an abnormally large response time, we have two options to learn more about the transaction. If it is a database query, for example, we can study its SQL statement by right-clicking on its marker and choosing to 'Describe Transaction'.

If its marker is enclosed in a purple square, NetData has noted some form of network abnormality such as a retransmission that occurred while the transaction was running. We investigate the details of the network's behaviour, and see where the transaction's time was spent, by right clicking and choosing to 'Plot Transaction Timing...'. This is probably the operation we perform most frequently, and it takes us to the timing chart, with or without packets as we choose.

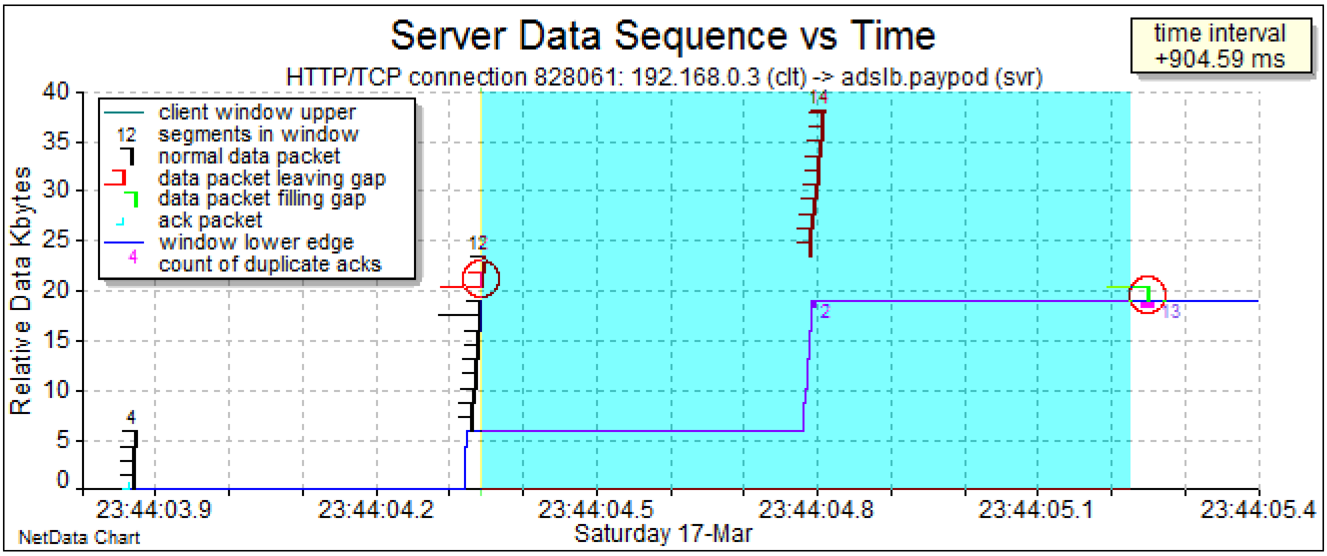
On the timing chart we often right click on a transaction bar and choose to 'Load All Packets & Trans of Connectn'. This shows all the transactions and packets leading up to the subject transaction, and the next step may be to request a data-flow chart of this connection, to investigate any packet flow issues.

Synchronising Charts

The performance and timing charts are regarded as *master* charts of equal rank. All the other charts and the supporting tables relate to one of these master charts. They provide quite different views of system behaviour, and it is often useful to ensure that both charts cover exactly the same time range. The **Timing** button above the performance chart generates a timing chart with the same time span, and the **Trans** button above the timing chart achieves the reverse, a performance chart with the same time span.

Time Interval Measurement

A time interval on a chart can be measured by dragging the cursor, with the Alt key down, from the start point to the end point. A blue rectangle is painted between the start point and the cursor, like the rectangle for zooming, and a pop-up in the chart's top right corner displays the value of the time interval. The cursor will jump to a nearby transaction or packet, thus giving the precise difference in the timestamps of two packets.



The red circles identify packets that were near the cursor when at the beginning and the end of the interval. The time interval displayed in the top-right corner refers to the difference in the timestamps of these packets.

Instant Chart Snapshots

To compare different views of system behaviour, snapshots of charts can be taken with either the **Save** button, to create an image file on disk, or the **Copy** button, to paste the image in a word-processor document via the clipboard. A third and quicker alternative is to type Alt-W which creates a copy of the chart's image in a new NetData window. The new window has no controls and can't be resized, but can be minimised for the tray.

Further Guidance

Training Videos

A comprehensive suite of training videos can be found at Phil Storey's YouTube channel:

https://www.youtube.com/c/networkdetective

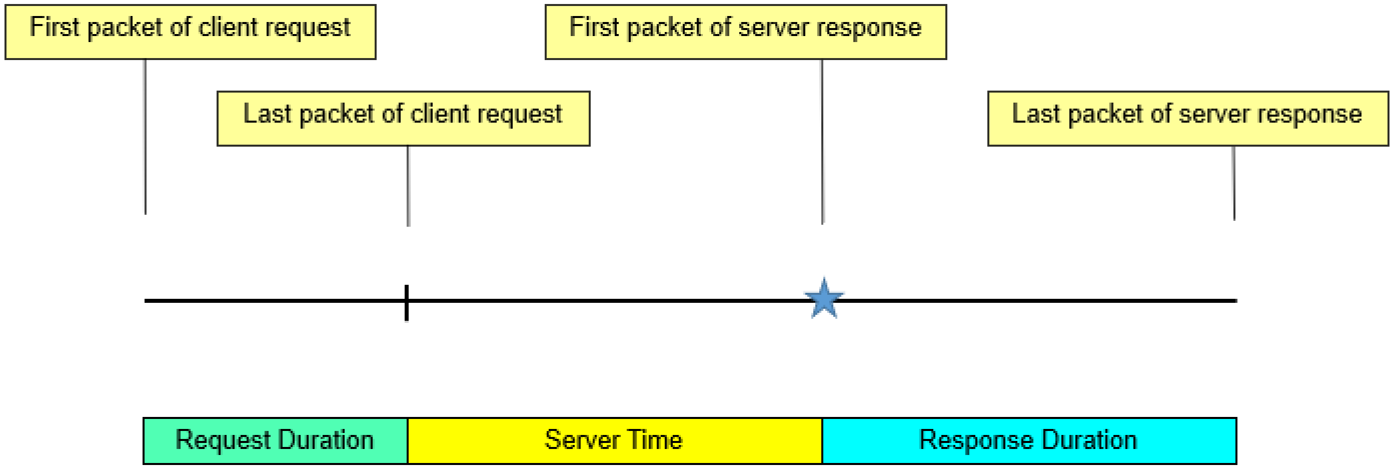
Transaction Representation

(Courtesy of Phil Storey)

Understanding Performance Chart Transaction Symbols

NetData measures and reproduces individual application-layer transactions. The Performance chart plots each transaction as a horizontal line that represents the overall transaction duration, adding symbols within the line to indicate timings of transaction components. The main symbol's colour reflects the transaction's server (or client) and its shape indicates transaction type.

The x-axis is time-of-day and the y-axis is transaction duration. Each transactions is plotted at a height that usually represents the transaction's overall duration (the full length of the horizontal line).

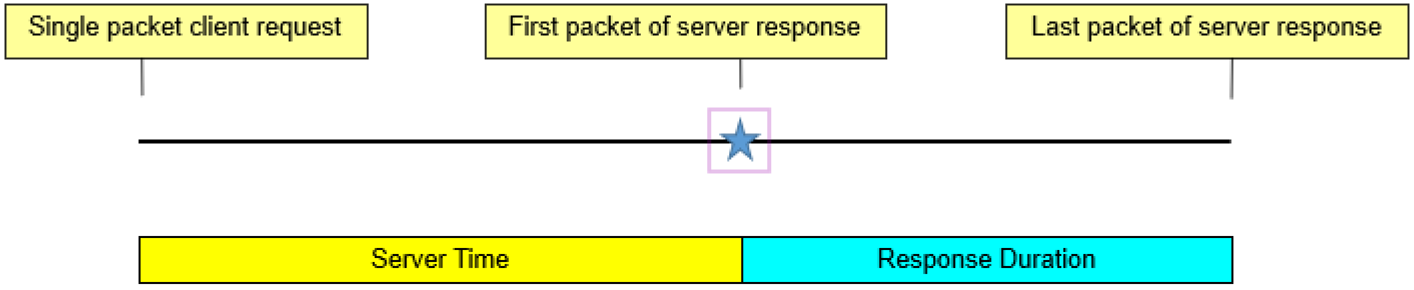


These green, yellow and blue colours are applied consistently to transaction bars on Timing and Waterfall charts to represent the same time components.

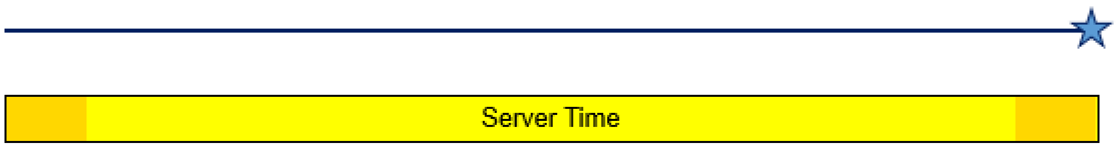
Understanding Performance Chart Transaction Symbols

When the client's request is small enough to be transferred in a single packet, there is no vertical tick and the whole length of the horizontal line before the coloured symbol represents 'Server Time'.

A pink square around the symbol indicates that the transaction has been affected by a network abnormality..



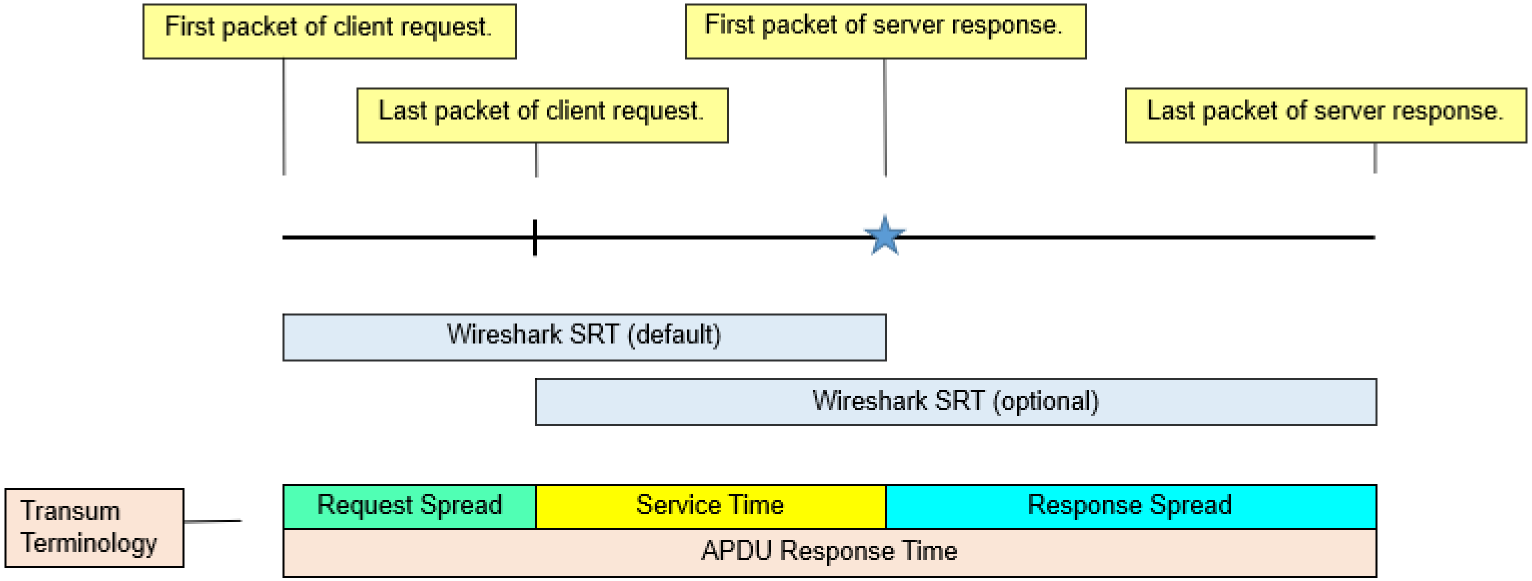
If the server response is also small (or relatively fast), then the coloured symbol will be at the very end of the line and the whole transaction duration appears to be 'Server Time'. In client-side captures, orange bars visualise propagation delay..

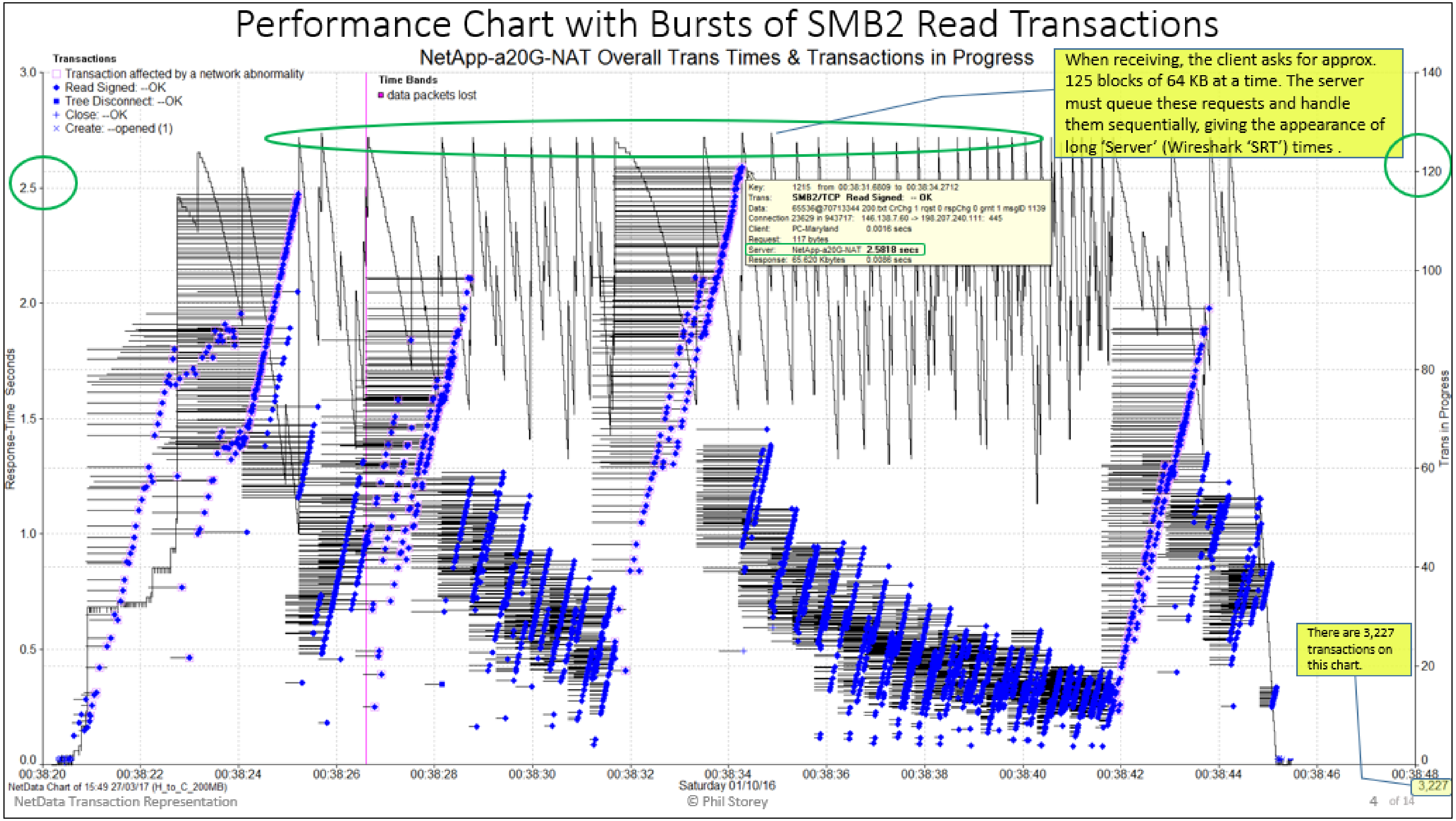


Comparison with Wireshark & Transum

Wireshark has two settings for measuring 'Service Response Times' (SRT) for several application protocols.

A newer feature, Transum (originally developed by Tribelab in the UK but now built into Wireshark), can also calculate the timing of request-, server-, response- and overall-transaction durations for some protocols.



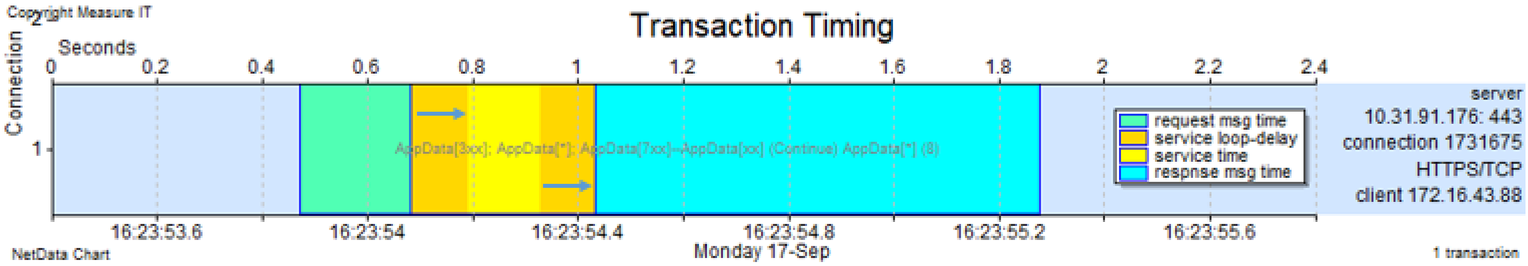


Performance Chart with Bursts of SMB2 Read Transactions
NetApp-a20G-NAT Overall Trans Times & Transactions in Progress
Transactions
Transaction affected by a network abnormality
Read Signed: --OK
Tree Disconnect: --OK
Close: --OK
Create: --opened (1)
Time Bands
data packets lost
When receiving, the client asks for approx. 125 blocks of 64 KB at a time. The server must queue these requests and handle them sequentially, giving the appearance of long 'Server' (Wireshark 'SRT') times .
There are 3,227 transactions on this chart.
Response-Time Seconds
Trans in Progress
Saturday 01/10/16
NetData Transaction Representation
© Phil Storey

Simple Transaction Timing Chart

The transaction–timing chart represents the durations of transaction phases with horizontal bars in the standard colours, drawn on pale-coloured bands dedicated to the activity of different connections. The connections are described in the chart's right-hand column and a floating box of legends tells what the colours represent.

In this simple example we have only one TCP connection and one transaction. Later we'll see many of each.

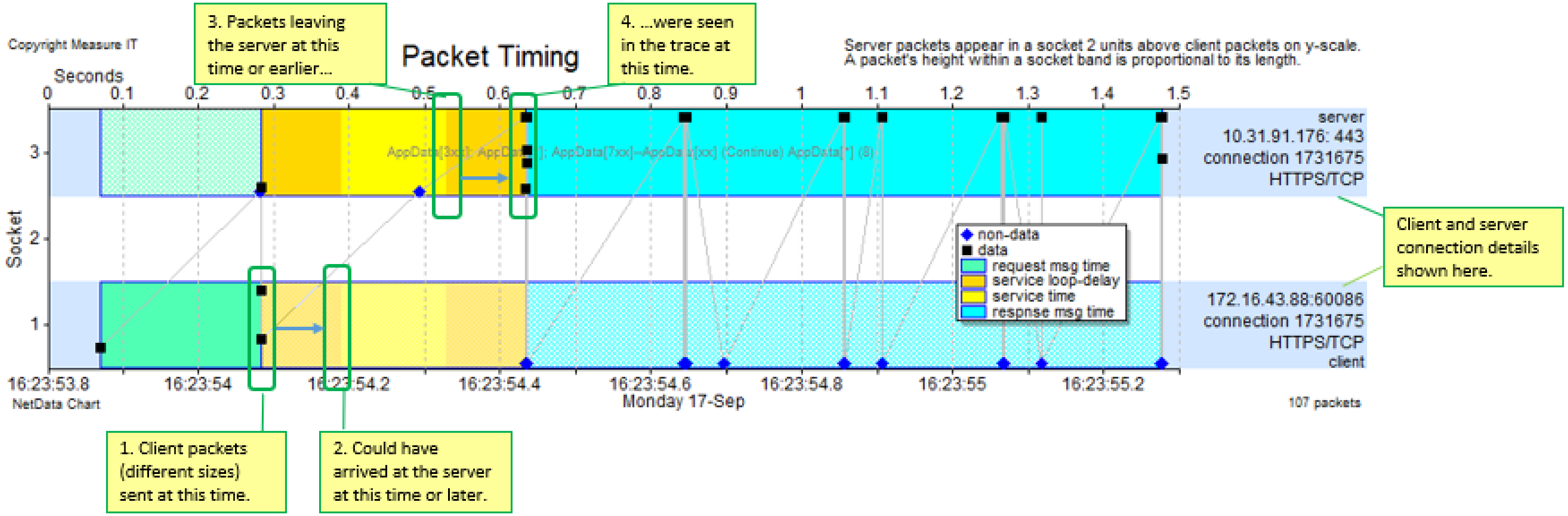


Optional orange bars indicate propagation delay associated with the service round-trip. The first orange bar represents the time for the last request packet to propagate from the sniffer to the server (arriving no earlier than the right-side of the bar); the second bar represents the time for the first response packet to propagate from the server(starting no later that the left-side of the bar) to the sniffer.

Simple Packet Timing Chart (One TCP Connection)

When packet markers are overlaid on transaction bars, the chart becomes a packet-timing chart. This is similar to a Wireshark Flow chart or "Ladder" diagram in other tools – but rotated onto its side. Packets are plotted at the times read from the capture file, making time intervals between packets easy to see.

There is no need for arrows to indicate packet direction because each connection band is split into two horizontal *socket* bands. Client packets are always plotted on the lower band and server packets always on the upper band. The height of a marker within its band is proportional to its length.

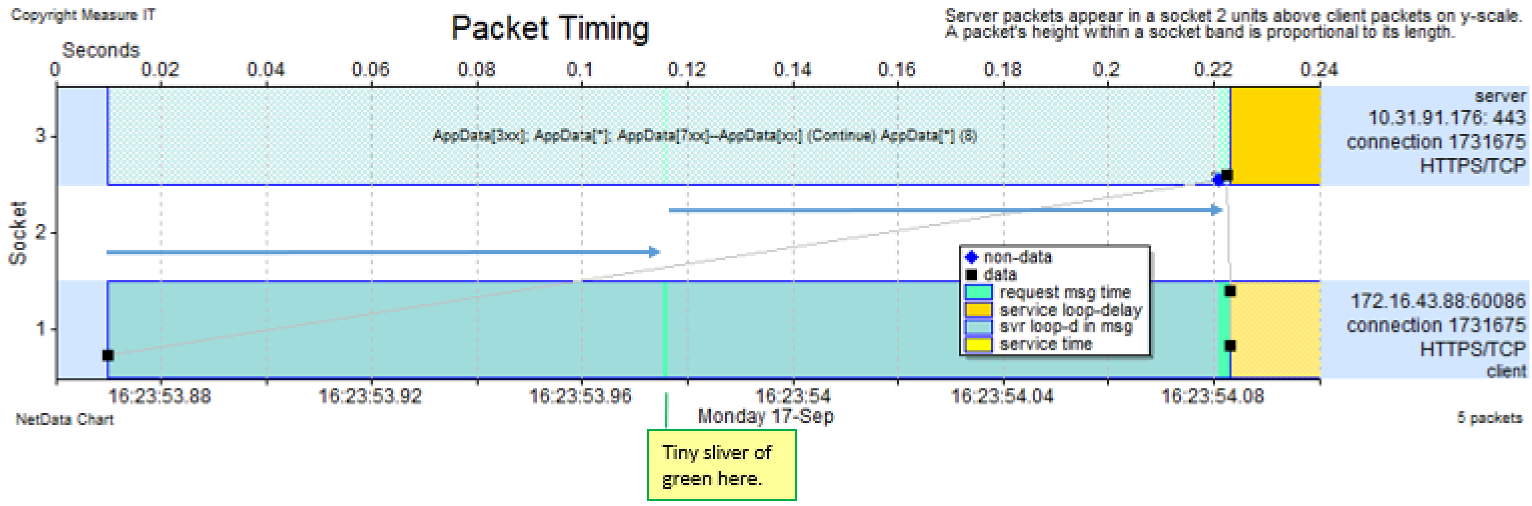


The optional grey lines link packets in chronological order. Different packet-marker shapes and colours (as indicated by the legend box) make it easy to identify different packet types: black squares for data packets, blue diamonds for acknowledgements. Many more are not shown here.

The server and response-message bars are drawn on the client band with paler colours, as is the request-message bar on the server band.

Propagation Delay

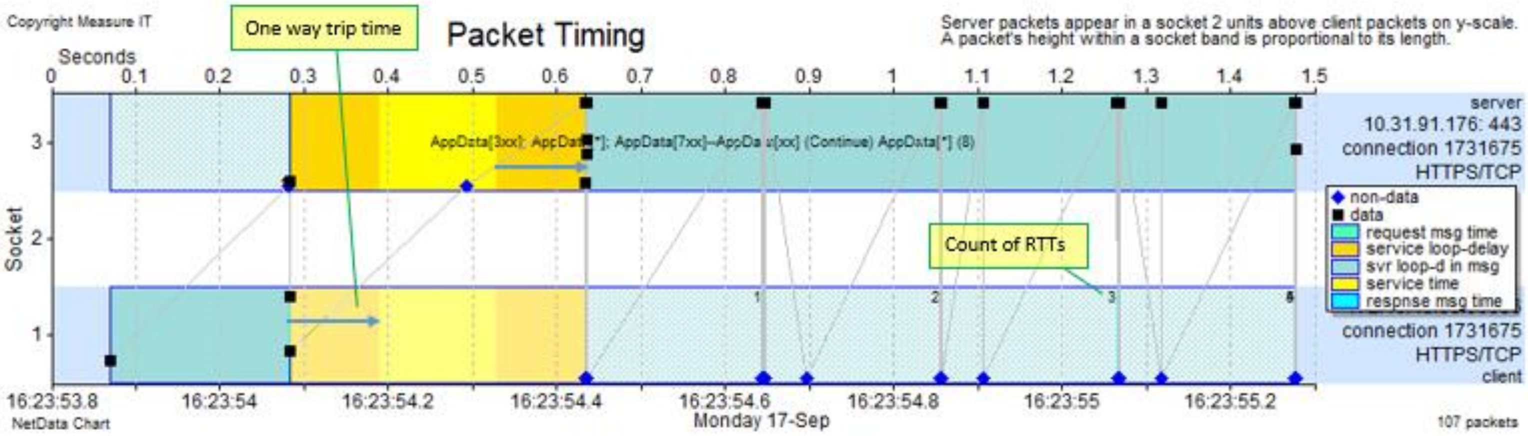
NetData can also indicate the time spent in signal propagation during request and response message transfers. During a transfer there are likely to be many concurrent activities; while some packets are being transmitted over different links, others might be queued in routers awaiting transmission, and many signals will be propagating. NetData identifies a 'critical path' through all the activities to avoid counting any time period twice. In effect it tracks the times for a single data packet to cross the network, say from server to client; then the times for the Ack packet prompted by that data packet to return across the network; and, if data is waiting, subsequent data packets as they are prompted when a responding Ack opens the window.



In this chart two grey-blue propagation bars are separated by a very short green bar. The first grey-blue bar represents the time for the first request packet to propagate from the sniffer to the server (arriving no earlier than the right-side of the bar); the second bar represents the time for the first Ack packet to propagate from the server (starting no later that the left-side of the bar) to the sniffer. In this case the client was waiting for a short data packet containing an HTTP 100-Continue message from the server before sending the remainder of the request.

Response Propagation Delay

Just as orange bars indicate propagation delay (half the minimum RTT) within the yellow server time areas, grey-blue bars indicate propagation delay during request and response message transfers.



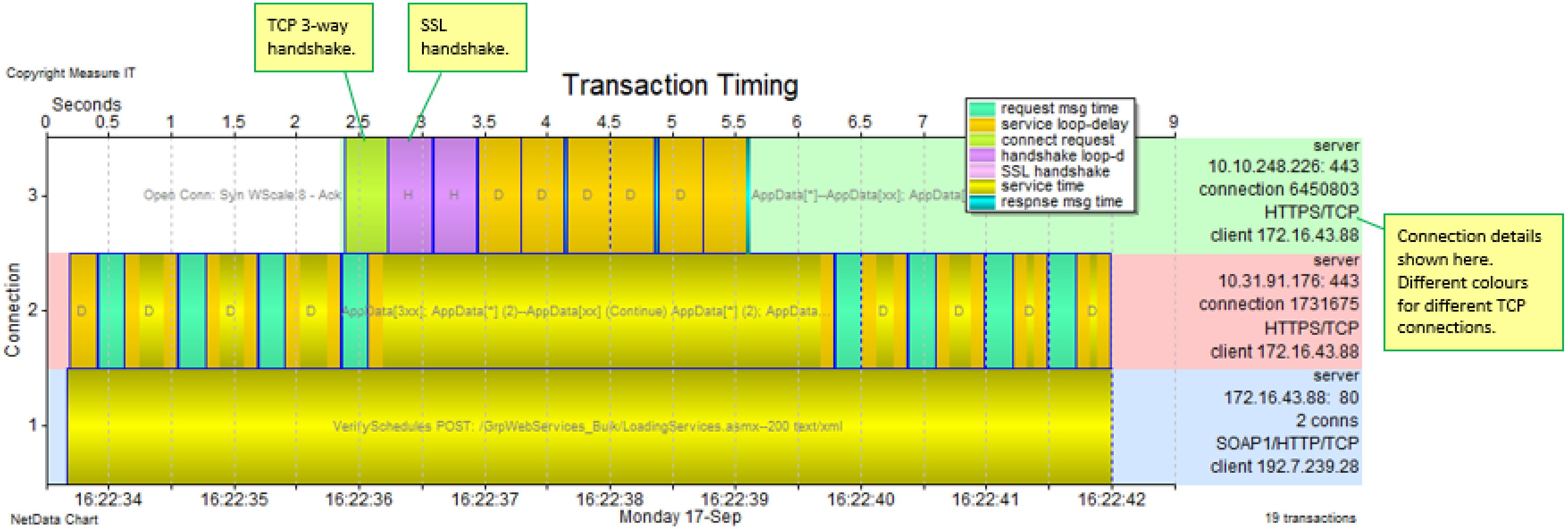
In this case the response message required four TCP round-trips, each trip starting with a data packet from the server and ending with an Ack from the client. At this scale it appears that all the message-transfer time was propagation time, with one loop-delay for the request message and four times the loop-delay for the response message. When the orange propagation time is added, it is seen that this transaction's overall response time included six times the path's loop-delay.

The individual TCP round-trips are counted with numbers displayed on the chart.

Activity on Many Connections

The timing chart scales to visualise the activity of possibly hundreds of connections with thousands of transactions (and optionally all their packets). In this case a single front-end SOAP transaction on the bottom band (connection 1) generated transactions on backend connections to two different servers (connection bands 2 and 3). The overall timing of the front-end transaction was determined by the timing of the nine back-end transactions on connection 2.

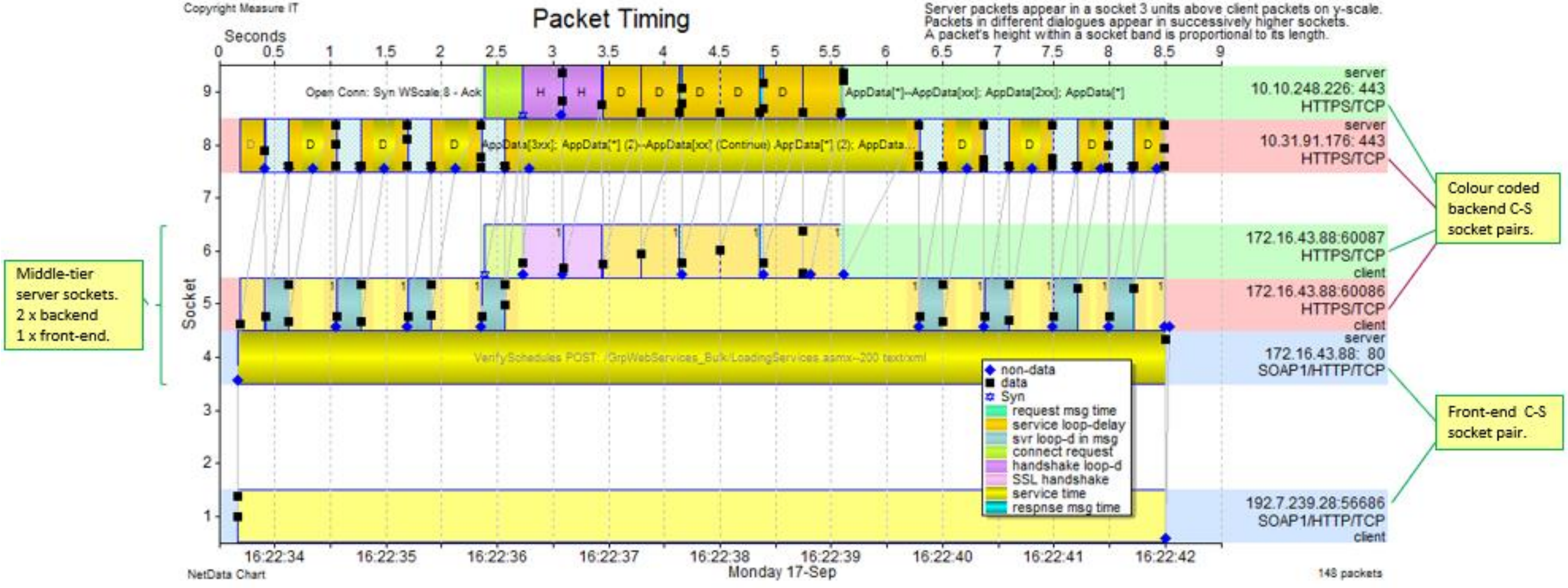
As always, the floating legend box describes the bar colours, including here a green bar for a TCP three-way handshake and two purple bars representing round-trips for an SSL handshake.



Characterising a Multi-tier Transaction

Adding packets to the chart splits each *connection* band into two (client and server) *socket* bands. The connection pairs are further separated to emphasise the multi-tier nature of configurations by placing front-end connections below backend connections where possible. In this chart the front-end connection pair (blue) is at the bottom and the corresponding backend connection pairs (red and green) appear at the top. All three bands clustered in the middle describe the streams of packets transmitted from three sockets of the middle-tier server.

Adding packets also allows NetData to count and display propagation delay in message transfers.

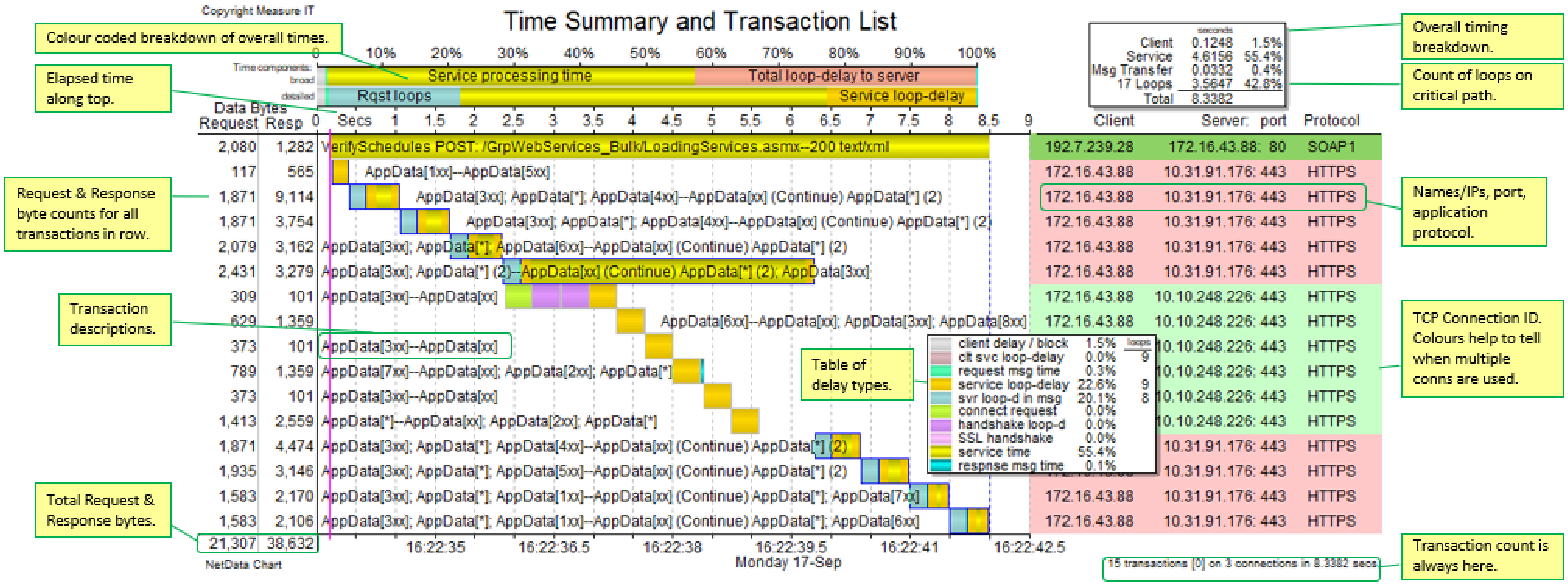


Interpreting Waterfall Charts

All the back-end activity generated by the single front-end transaction can also be characterised on a waterfall chart, with one application transaction on each row. The top row is the same front-end transaction as before and subsequent rows describe the various backend transactions.

Time summary bars and their table above the waterfall tells that backend propagation delay accounted for 42% of the overall response time.

NetData also identifies a 'critical path' through the various system activities and counts round-trip times (loops) along this path. The absence of blue borders to the transaction bars on the green connection indicates that their activity was not counted on the critical path.



Three Different Server Types

In this example, the overall function took 1.33 seconds.
On this chart each row contains multiple transactions of the same general category.
It is easy to see when server time occurred (yellow) and when response network data delivery occurred (blue).