Why NetData?

Contents

Why Analyse Network Traffic?

> *Protocol analysis remains one of the most rewarding yet challenging aspects of troubleshooting applications, networks and server performance…. Fifteen or twenty years ago, deep packet analysis was the way to solve complex performance issues and it still is today.*
>
> Rob Webb, CA Technologies, February 27, 2018

NetData analyses network traffic and undertook its first consulting assignment twenty years ago, to diagnose a long-standing problem that blocked the rollout of a nation-wide application for the Department of Health. Now, after further continuous development, NetData diagnoses as many application performance problems as it does network problems.

While all traffic analysers can differentiate between server processing and network delays, only NetData demonstrates an ability to view, for example, the concurrent activity of multiple application threads in a backend network. In combination with NetData's thorough descriptions of transaction contents, and the precise timestamps on all the packets, these highly detailed pictures provide unique insights into how applications work.

It doesn't matter where faults lie – in application code, a server resource, a TCP driver, a network device, or in the interaction between, say, a load balancer and a server – if a transaction is slow or fails, it is bound to appear on a NetData chart. That provides a starting point for characterising the problem and drilling further, perhaps with captures from other parts of the system or other times.

Network traffic can be captured with virtually no footprint on the system, or more conveniently by sniffers already installed on servers. It provides the only coherent, objective and highly detailed view of a system's behaviour over time, and this is why NetData is also an ideal tool for triage and narrowing the problem domain.

Can Other Tools Do the Job?

NetData complements enterprise-wide system management tools such as those provided by ExtraHop, NetScout, Tivoli and Riverbed, adding analytical power when the focus moves to a specific part of the network or an application, and packet analysis is warranted. But this is not to say that

NetData is effective only in harness with a monitoring system. Indeed, when a performance problem arises, there is always an effective path to a diagnosis that starts with the capture of network traffic containing evidence of the problem – perhaps a slow or failed transaction. The process can continue with NetData alone to characterise the problem and guide subject-matter experts to the solution. The traffic-capture step is of course much quicker if a continuous monitoring system can ‘go back in time’ to retrieve relevant traffic, and it is always helpful to corroborate findings with other tools.

Many other tools analyse network packets, most produce charts, and nearly all claim to find root causes faster than the others. But their sample charts and diagnoses are instructive. Few of their charts have more than a thousand data points, whereas NetData’s charts can present hundreds of thousands of data points showing, for example, the response times of individual transactions, without any averaging. These charts permit many different overlays to explore correlations with other aspects of system behaviour.

While NetData can display trends in averages, they are rarely needed when investigating system behaviour. Averaging hides huge amounts of diagnostic information, and it is the patterns of individual response times, overlaid with graphs of transaction queue lengths and numbers of concurrent connections that can reveal the existence of database blockages, saturated processing power, or overloaded disk channels; reveal stress in a sending host, a receiving host or network equipment; and reveal forms of unhealthy behaviour still to be understood by the troubleshooter.

A system needs its dashboard fuel gauges – to show, for example, CPU, bandwidth and memory utilisation – but they are no substitute for visualising how a system behaves over time, like a slow-motion movie. There is consummate detail in NetData’s charts and their timescales can range from less than a millisecond to many hours.

Some tools calculate only an average response time across all the types of transactions handled by a server. A major benefit of NetData's detailed characterisation of transactions is the automatic definition of appropriate transaction *types*. The types are presented in a tree which allows all the transactions of a particular type, or group of types, to be loaded for display on a performance chart. Such charts tell whether all similar transactions are slow, or their performance depends on other transactions in progress.

The ability to view system behaviour, formulate hypotheses and test them by generating special types of charts is far removed from the root-cause analysis claimed by other tools. The 'superior' root-cause analysis of one tool was illustrated by its ability to identify a 404-status code in a web response. Such tools are mere toys alongside NetData which tests hypotheses and answers questions that appear not to have occurred to the makers of other tools.

NetData has maintained a reputation for cracking the most complex performance problems over 20 years, and in that light has frequently been engaged in consulting assignments only when all other diagnostic tools and avenues have been exhausted. When a problem is resolved it is often concluded that no other traffic analyser could have done the job.

Objectives

NetData's strength derives from its unique graphics and its exhaustive database. Its simple objective is to extract as much diagnostic information as possible from network traffic. It was designed to be thorough, to reveal all signs of unhealthy behaviour, whereas other tools are designed primarily to display information on dashboards in real time. NetData doesn't capture traffic but reads the capture files of all the sniffers Measure IT has encountered – more than 25 capture-file formats so far.

When NetData scans all the packets in a sequence of capture files it carries forward extensive state information not only about each connection but also

its secondary connections such as opened files and database cursors. This is particularly important in analysing database traffic when in one round-trip the client submits a Select statement with bound variables, and much later might re-execute the query with different search targets – different values for the bound variables. NetData locates the earlier transaction, to retrieve names for the variables and metadata for the table columns to fully reconstruct the result set in every query.

NetData can display the SQL statement for every query, and the tables of metadata and the result sets in every response. It does this with all the major database protocols including MySQL, SAS, Teradata, Interbase, Sybase, Microsoft SQL Server (TDS), IBM DB2 (DRDA) and Oracle SQL*net (TNS). Oracle has been careful to keep its protocol specifications out of the public domain, and to the best of our knowledge no other tool can decode much more than the 8-byte headers of Oracle packets. Some tools manage to display SQL statements only by scanning message contents for SQL verbs.

With NetData the analyst can contemplate a much larger list of failure mechanisms that are difficult if not impossible to check with any other tool, such as:

- Ceilings for the numbers of threads, connections, router buffers
- Dangerously small recycle times of ephemeral ports
- Polling delays
- Limited send-buffer space
- Inappropriate, obscure TCP parameters
- Queues for different transaction types
- Transaction redundancies
- Database abuse with unnecessary round-trips
- Database blockages; missing table indexes
- Single-threading blockages
- Short-term bottlenecks at different points in system
- Obscure bugs in protocol drivers
- WAN accelerator faults
- Application coding inefficiencies

- Absence of buffering for application output
- Changes in transaction mix
- Mismatched timeouts across system
- Connection-pool misconfiguration
- Insufficient packet-queue buffer space
- Inappropriate packet-shaper and –policer parameters

The following sections describe just some of NetData's more powerful and unique capabilities. Supporting this summary are two separate documents:

1. *NetData Pro Charting Guide* – to introduce analysis and chart types.
2. *NetData Overview* – illustrating many unique types of analysis.

Multi-Tier Analysis

Relating front-end to backend transactions usually requires a software agent to track the activity of application threads, but NetData needs only traffic captures and relates transactions with a pattern-matching technique, finding bursts of backend activity that coincide with the time spans of front-end transactions. This technique exploits the single-threaded nature of transaction handlers that activate only one backend connection at a time. A timing chart displays families of backend transactions alongside their front-end parents, allowing the analyst to verify or edit relationships, and to see where time was spent.

Waterfall Charts and Database Abuse

NetData can generate a waterfall chart for any group of transactions, no matter how many servers and application protocols may be involved. One of the four different types of waterfall chart dedicates rows not to individual *transactions* but to individual transaction *types*. If the waterfall covers all the database transactions generated by a single user transaction, and some query types are executed many times, the chart characterises the loops within the

application program and reveals opportunities for major reductions in database trips.

Redundant or inefficient database use is now quite common and often wastes many seconds. No other traffic analysis tool and very few system monitoring tools point so clearly to this cause of large response times.

Correlating Related Captures

Concurrent captures might be taken from different points in the network to investigate abnormal transit times of different network segments or of equipment such as firewalls. The challenge then is twofold: to find all the packets that are common to different captures; and to correct their timestamps for differences in sniffer clock settings. Only then can timestamps be compared in order to measure transit times.

NetData has a tool that reliably finds all the matching packets, even when addresses and TCP sequence numbers are translated. Furthermore, NetData can create a *super* analysis project that points to any number of related captures, and each of those captures may comprise any number of contiguous capture files up to 100 GB or more. In a single command NetData will analyse all the related captures simultaneously – using a different processor core for each capture – and when finished will find all the matching packets. It then processes pairs of timestamps in its linear regression engine to determine not only differences in clock settings but also differences in clock speeds; it adjusts timestamps so that all times are expressed according to a common clock, and measures the transit times of all the packets. Correcting clock speed is crucial when measuring transit times with a sniffer's resolution of microseconds, and NetData is probably the only tool that corrects both clock setting and speed.

Charts of transit times can colour their markers according to a variety of packet attributes such as WiFi or Bluetooth signal strength, connection ID, or

their MAC addresses. In the latter case the chart might indicate which packets are taking different paths across the network, and which paths are more congested.

Plotting Queue Lengths

To find even the briefest of performance slowdowns in the largest captures NetData has *activity-overview* functions to chart the high-water marks of transaction-queue lengths, packet-queue lengths, and numbers of concurrent connections with individual servers. Very few if any tools plot queue lengths and an event can be found only if it causes a noticeable change in traffic volume. Only the most obvious failures do so.

Relative Transit Times of VoIP and TCP Traffic

Jitter is a major cause of poor quality in VoIP calls but the standard measure for jitter involves so much smoothing that it provides no clue to jitter causes. NetData helps to identify likely causes by avoiding any form of averaging and plotting the transit times of individual packets in RTP streams. NetData documentation shows how patterns in these charts can indicate an overloaded firewall, frequent swapping between alternative paths, misconfigured de-jitter buffers and significant differences in codec clock speeds.

NetData is unique in being able to measure relative transit times, and it does so without the need for captures at both ends of the path. It compares the sniffer's timestamp with the codec's timestamp conveyed in each packet, but only after processing the pairs of timestamps in its regression engine to adjust for any difference in clock speeds.

If TCP headers include timestamps, NetData can perform a similar analysis, comparing sniffer and TCP timestamps to measure transit times from the sender to the sniffer.

Microbursts and Queue Modelling

Microbursts – short, rapid bursts of packets – have attracted much attention in networks handling securities trading traffic because they might lead to packet losses when a queue buffer overflows, and the ensuing delays give a trading advantage to competitors. However, they deserve attention in all networks because queue overflows are probably the most common cause of packet loss and degraded throughput in today's networks.

It appears that all tools, including specialist tools for investigating microbursts, attempt to reveal the possibility of microbursts by plotting link utilisation over very-small time intervals of a millisecond or less, and flagging intervals whose utilisation exceeds a specified threshold. However, there need not be any correlation between utilisation and the occurrence of microbursts. It is quite possible that a well-paced data flow can achieve a utilisation of 100% over very long periods without putting any pressure on buffer space. Conventional tools generate many false positives.

NetData takes a more appropriate and direct course to what matters. It first runs all the recorded packets through its packet-loss inference engine, to identify those packets that were lost after being seen by the sniffer, presumably when a downstream queue overflowed its buffer. It then models the behaviour of such a queue when handling all the packets destined to use it, and if a chart of queue length shows that packets were lost only when the length reached some peak, then the validity of the modelled link speed is confirmed, the peak indicates the buffer size, and the cause of packet loss is understood. Comparison of the measured round-trip times with the modelled queue-waiting times corroborates the model's validity, and other overlays on the queue-length chart show the detailed composition of the troublesome microbursts – the connections to which the packets belong, and the flow-control rules governing their transmission.

Traffic flows are often throttled artificially, to limit bandwidth use to contracted Peak and Committed rates. Packets are lost when an application attempts to exceed these rates, and the resulting effect on performance can be debilitating and difficult to diagnose. Network support staff may not be aware of the configuration of packet shaping and policing systems in their network. However, NetData can model the behaviour of shapers and policers, and when charts show that packets were lost in consistent circumstances we not only learn the cause of the performance degradation but also the parameters of the responsible shaper or policer.

Integrating Fiddler and HttpWatch Output

Now that most web traffic is encrypted, and servers run in a cloud, it is increasingly difficult to diagnose performance problems. NetData measures the response times of HTTPS traffic and characterises transactions according to the patterns of the SSL records that compose their request and response messages. The missing piece of the diagnostic picture is message content ‘in clear’, before encryption, and that can be provided with two other tools: with Fiddler that acts in the role of a proxy server to intercept and re-encrypt the traffic; or with HttpWatch which runs in the client browser to record messages before encryption.

Fiddler necessarily affects the way the network is used, and its message timestamps have a low resolution. HttpWatch is more accurate, but it doesn’t see network packets and sometimes provides a quite misleading picture of network and server performance. NetData can import all the information in Fiddler and HttpWatch files, recording their views of transaction performance and content in its database. NetData’s performance, timing and waterfall charts can then present all the information side-by-side, combining the sniffer’s accurate view of network behaviour with, say, HttpWatch’s description of message contents and significant browser events.

Modelling Network Changes

When workstations or data centres are moved, and networks are reconfigured, there is a need to determine the likely impact on transaction response times. QuickPredict in Riverbed's Transaction Analyzer is one of the few tools that purport to show how a transaction's response time varies with changes in link bandwidth, utilisation and propagation delay, but its mathematical basis and results are wrong in all but the simplest cases. Most calculations are distorted by an inability to properly handle concurrent activities in the progress of a transaction. Parallel activities occur, for example, in the rendering of a single web page that typically needs 100 or more files, from multiple servers, using multiple connections.

Adding together all the server processing times and all the message-transfer times yields a sum of component times greater than the overall elapsed time, and some tools hide this gross error by simply applying a scale factor (a 'squeeze') to achieve the correct total, but this too is mathematical nonsense.

A similar problem arises when measuring the different categories of time delay involved in transferring a message. Some tools calculate the time to transmit *all* the packets, and accumulate the congestion delays of *all* the packets, but this also produces nonsense because it ignores the overlapping of activities: while some packets are being transmitted, others are waiting in buffers for their turn to transmit or waiting for an ack packet to propagate across the network and open the transmit window.

NetData avoids double counting delay components by choosing a single 'critical' path through all the activities – like the critical path of a project-task diagram – and characterises how time is spent along that path. It does this not only for application round-trips but also for the components of network delay. The result also provides an accurate count of the true number of 'turns' or 'loops' – the number of times that the network's propagation delay is incurred on the critical path. The timing chart uses different colour bars to

identify periods of propagation delay, for application round-trips (orange) and during message transfers (blue-grey). The waterfall chart identifies the critical path and it can be changed manually.

NetData determines the packets incurring transmission time (for bit serialisation) on the critical path and can therefore model accurately the effect of a change in link speed. But the major component of response time over modern high-speed networks is usually propagation delay, and with its accurate loop count NetData is probably the only tool that models the effect of propagation delay accurately. This modelling is illustrated in the charting guide.

Where and When is NetData Best Used?

NetData plays a valuable role during application development, helping to debug code and identifying any inefficient use of the network. It provides a valuable health check before an application leaves the laboratory and undergoes expensive load testing – there is little point in performing load tests if the application is imposing too many network round-trips, or flow-control parameters are not set correctly.

Traffic analysis in the development lab provides useful data for capacity planning.

During load testing NetData confirms that load generators are simulating users correctly and are themselves not overloaded. It plays a vital role in identifying any bottlenecks encountered during tests.

NetData's traditional role, of course, is in diagnosing production performance issues. However, if there is adequate spare time, NetData should be used to perform regular health checks on the production system, and especially after any significant system change. Network support staff are rarely able to check the correctness of a configuration change beyond confirming that a few transactions can be handled successfully. NetData, however, reveals many

forms of unhealthy behaviour well before they grow into serious performance problems.

Is NetData Easy to Use?

The graphical display of virtually all analysis results makes NetData easy to use, and its analytical power ensures that it can achieve results in fewer steps than other tools – if they can be achieved at all.

A first-time user could be confused by the many options for the analysis phase and for formatting every type of chart, but this simply reflects the number of techniques needed to investigate the infinite variety of problems that can afflict IT systems. To mitigate this confusion all controls are initialised with their most useful settings, making it always safest to take the default path when a dialogue box pops up.

The first-time user may also be confused by the first charts to appear after an analysis, but this is deliberate. They are designed to introduce most of the overlays that can appear on a chart, relying on the user to open the chart's format-control window and hide the unwanted detail. An experienced user will load only the database records that are wanted for a chart.

Measure IT encourages feedback and suggestions from users for easier ways to generate charts, and for new types of analyses and charts. We use NetData every day and take every opportunity to make it easier to use.

NetData's user documentation runs to many hundreds of pages, but experience is the best guide when interpreting charts, developing hypotheses to explain system behaviour, and choosing techniques to test them. That is why users are encouraged to discuss problems and charts with a Measure IT consultant.

Simpler Capture Requirements

Some tools focus on a specific task or transaction, not just packets, but they require the analyst to provide capture files that have no extraneous traffic. NetData imposes no such requirement. Rather, a NetData analyst is encouraged to capture all the traffic because, until a problem is fully understood, any packet might be relevant. Once analysed, focusing on the interesting traffic is easy. Any object – client, server, connection, dialogue, transaction, packet, network event or application error – can be selected on any chart or table; then its related records can be loaded from the database to populate new charts. Another set of filters apply to the loaded chart data, to further customize charts and zoom into the interesting patterns.

To measure transit times and congestion delays some tools require traffic to be captured at both ends of the network, but there is no such imperative for NetData. If it has a capture from only one point in the network, NetData will measure and plot the round-trip times of all acknowledged packets, and the resulting charts, overlaid with graphs of data sequence and bytes-in-flight, provide a thorough view of any congestion during message transfers. It doesn't matter whether the traffic is captured at the client, server, or some mid-point; NetData will plot the bytes-in-flight as seen by the sending host. This is another unique capability that reveals the rules by which data flow is governed, and the many possible causes of low throughput.

Resources

Ask for a demonstration analysis of a current problem, a trial licence subscription to NetData Pro, or download the free NetData Lite:

```
Domain name:  measureit.serveftp.net
Browser URL:  ftp://measureit.serveftp.net/
User name:    NetDataLite
password:     visual!ser
```


Contact:

Bob Brownell

Director, Measure IT

Bob@netdata-pro.com