



Network Trace Analysis Service

Problem Investigation: FTB Application Performance Problem

for

Investment Fund Manager

24th January 2018

Analysis by:

Darrin Roach, Performance & Stability Engineer, Advance7



Advance Seven Limited
Endeavour House
Coopers End Road
Stansted
Essex CM24 1SJ
www.advance7.com

This is an example of a typical network trace analysis report. Although the report has been abridged and certain facts redacted, it's based on a real project. The findings, conclusions and recommendations presented here are a true representation of those in the original report.

Contents

Executive Summary	4
Background	4
Findings Overview	4
Next Steps	4
Problem Description	5
Data Uploaded	5
Analysis Method	6
Findings	7
Database Response Times	7
Database Transaction Rate	10
Excessive TCP Retransmissions	11
Other Points of Note	12
Workaround	12
Conclusions	13
Recommendations	14

Executive Summary

Background

The Financial Trade Booking (FTB) application is failing to meet performance targets, and this is evident from warnings in the application log relating to slow database performance. There were no database performance warnings in the application log prior to the virtualised database server being vMotion'd from the VMware ESX server DEFESXP124 to the server ABCESXP135.

The Database Team has checked the FTB database and reported that they cannot find a problem. The suspicion is that there is a problem with the VMware ESX server ABCESXP135.

Findings Overview

Analysis of network packet data collected at the time of the problem shows that:

1. Database performance is satisfactory.
2. The performance problem is being caused by excessive packet loss on the TCP sessions that make up the pool of connections between the FTB application server and its database server.

Next Steps

We propose that the next steps are:

1. This report is share with stakeholders.
2. The *Findings* on page 7 and *Conclusions* on page 13 are discussed with the relevant teams.
3. Outstanding questions are discussed with Advance7 during the debrief conference call.
4. The recommendations in this report are actioned – see *Recommendations* on page 14.

Problem Description

The purpose of the application was described like this:

The application processes financial trade bookings.

The problem was described like this:

The Financial Trade Booking (FTB) application is failing to meet performance targets, and this is evident from warnings in the application log relating to slow database performance.

This differs from normal behaviour like this:

There were no database performance warnings in the log prior to the virtualised database server being vMotion'd from DEFESXP124 to ABCESXP135.

Data Uploaded

Network packet trace data was uploaded via the NTAS portal:

Trace File Name	Capture Location
app_*	A series of 6 PCAP-NG files captured using Wireshark dumpcap running in a VM via a vSwitch promiscuous port group that included the FTB application server vNIC. Collected on 19 th January.
db_*	A series of 32 PCAP-NG files captured on the FTB database server using Wireshark dumpcap. Collected on 19 th January.
app_mon_*	A series of 4 PCAP-NG files captured using Wireshark dumpcap in a VM via a vSwitch promiscuous port group that included the Application Server vNIC. Collected on 22 nd January.
db_mon_*	A series of 11 PCAP-NG files captured on the database server running on DEFESXP124. Collected on 22 nd January.

Other data provided was:

File Name	Containing
None	

Analysis Method

Analysis focused on flows between the FTB application and database servers.

The FTB application logs reported a high rate of slow database response times around 10:30 am on Friday 19th January, and so the analysis was focused on the period 10:20 to 10:30. During this time, the database server was being hosted on ABCESXP135.

The analysis was repeated using data collected on Monday 22nd January to check performance after the database server was moved back to DEFESXP124.

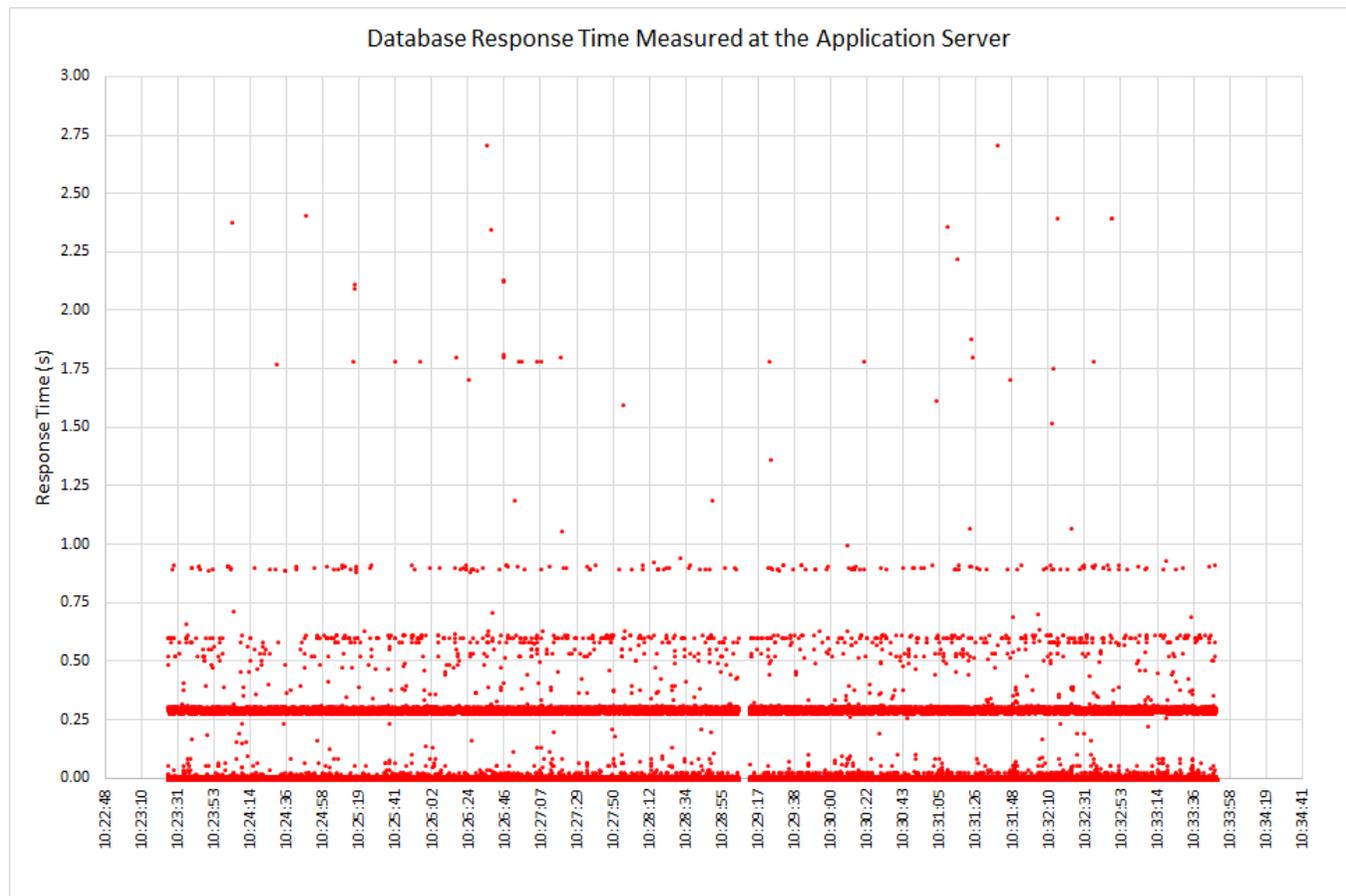
The data was pre-processed with a combination of tools to provide the Performance & Stability Engineer (PSE) with a roadmap to the data. The PSE then conducted detailed analysis, again, with a range of tools. The PSE will often refer to colleagues to help interpret the data.

The draft version of this report was then shared with a colleague for peer review.

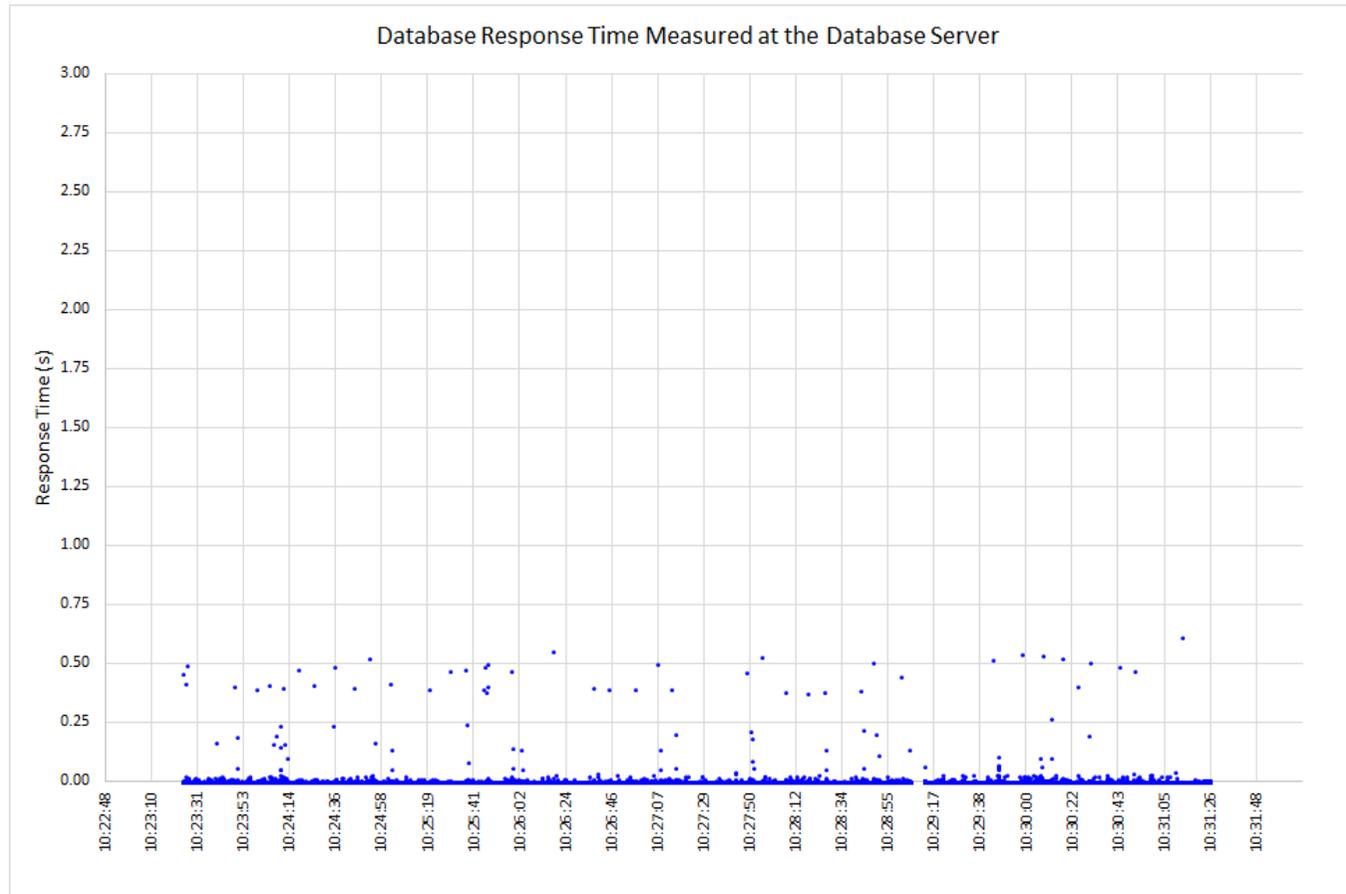
Findings

Database Response Times

The following chart shows the database response times measured by processing network trace data that was captured at the interface to the application server. We can see that there is spread of response times from a few milliseconds to 2.7 seconds.

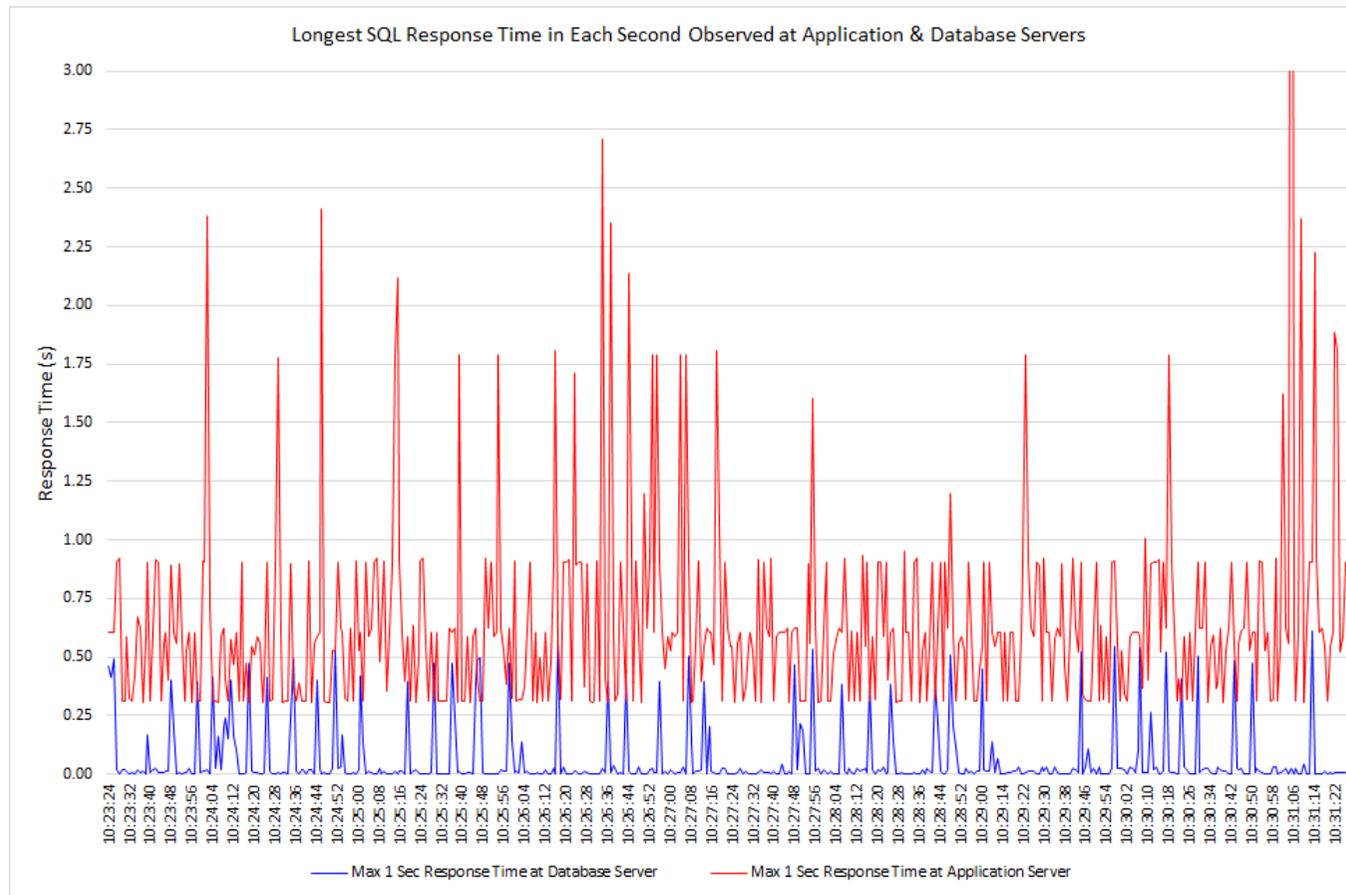


Variation in the database response times doesn't necessarily represent a problem. However, if we view the same response times from the perspective of the database server we can see there is a considerable difference.



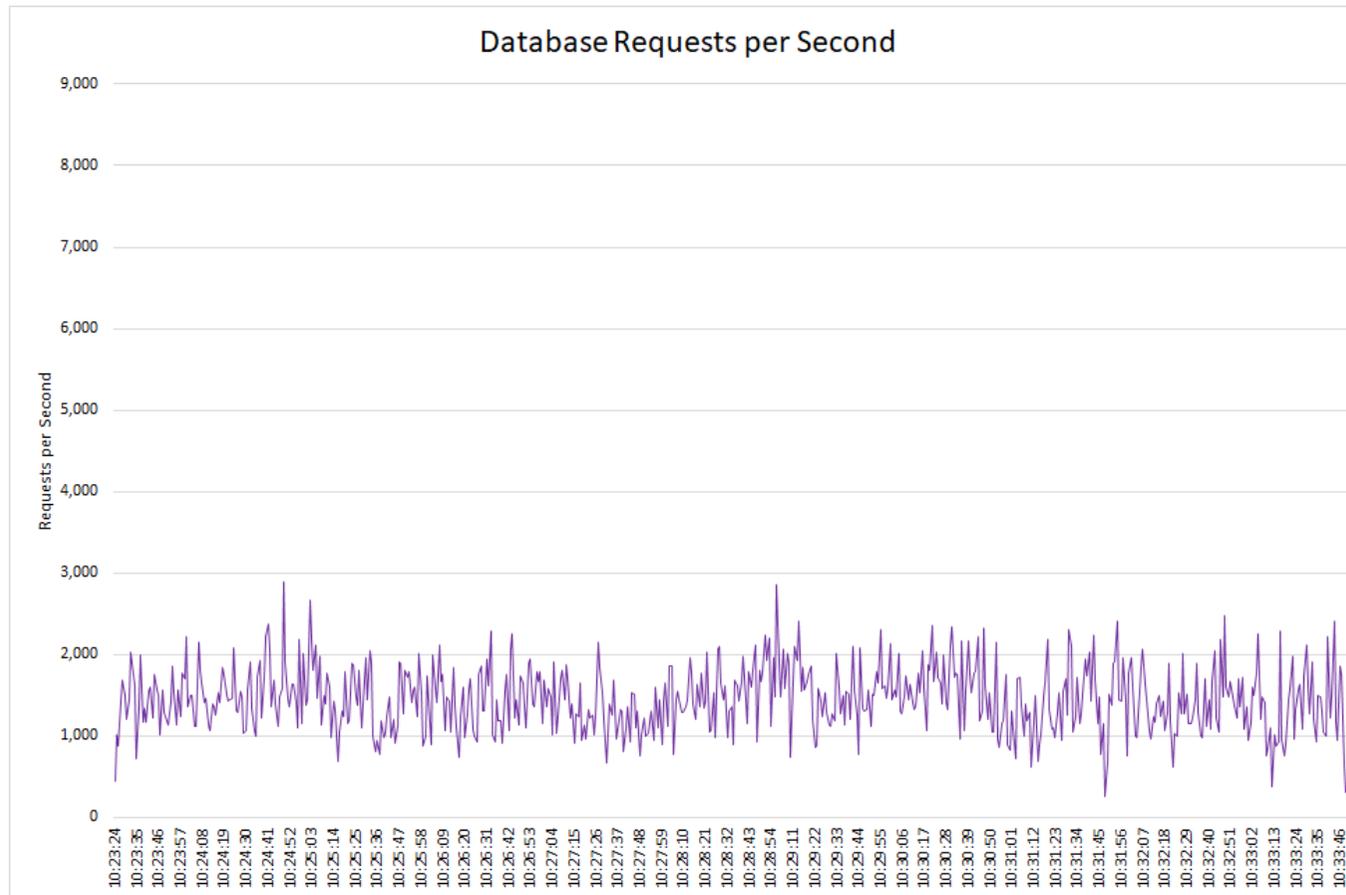
As we can see in the chart above, the database response times extracted from the packet data captured on the database server shows that the database appears to be performing much better than is apparent in the previous graph.

If we then compare the maximum SQL response times experienced at the application server and at the database server, an issue is immediately evident.



Although time synchronisation between the trace captured at the application server and that captured at the database will not be perfect, it is quite obvious that something between the two is introducing a delay.

Database Transaction Rate



The SQL transaction rate over the measured period is quite consistent.

Excessive TCP Retransmissions

Analysis of the traffic on the TCP connections that made up the connection pool between the application server and the database showed:

- The number of concurrent TCP connections between the application and database servers was 55
- A very high retransmission rate was observed in the TCP connections between the two servers only and was measured at 5.6%
- Retransmissions were triggered by intermittent packet loss in both directions between the two servers and across all 55 TCP connections
- The excessive packet loss occurred about once every 1 to 3 seconds on every TCP connection
- The time lost due to recovery of dropped packets frequently exceeded one second
- The effect of the packet loss was to extend the time taken for:
 - The request from the application server to arrive at the database server
 - The response from the database server to arrive at the applications server

The screen shot below shows examples of retransmissions in packets captured on the database server.

No.	Time	Delta	Source	Destination	Protocol	Src Port	Dst Port	Length	Info
80922	04:32:33.35629	0.000276	10.33.	10.33.	TDS	1433	50924	1041	Response
80925	04:32:33.356925	0.000396	10.33.	10.33.	TCP	50924	1433	99	50924 → 1433 [PSH, ACK] Seq=16453856 Ack=1689267774 Win=513
80927	04:32:33.357164	0.000239	10.33.	10.33.	TDS	1433	50924	80	Response
80929	04:32:33.357611	0.000447	10.33.	10.33.	TCP	50924	1433	1514	50924 → 1433 [ACK] Seq=16453901 Ack=1689267800 Win=513 Len=
80930	04:32:33.357613	0.000002	10.33.	10.33.	TCP	50924	1433	860	50924 → 1433 [PSH, ACK] Seq=16455361 Ack=1689267800 Win=513
80931	04:32:33.357668	0.000055	10.33.	10.33.	TCP	1433	50924	54	1433 → 50924 [ACK] Seq=1689267800 Ack=16456167 Win=256 Len=
80934	04:32:33.357911	0.000243	10.33.	10.33.	TDS	1433	50924	1041	Response
80939	04:32:33.358544	0.000633	10.33.	10.33.	TCP	50924	1433	99	50924 → 1433 [PSH, ACK] Seq=16456167 Ack=1689268787 Win=509
80940	04:32:33.358676	0.000132	10.33.	10.33.	TDS	1433	50924	80	Response
80943	04:32:33.359277	0.000601	10.33.	10.33.	TCP	50924	1433	1514	50924 → 1433 [ACK] Seq=16456212 Ack=1689268813 Win=509 Len=
80944	04:32:33.359278	0.000001	10.33.	10.33.	TCP	50924	1433	860	50924 → 1433 [PSH, ACK] Seq=16457672 Ack=1689268813 Win=509
80945	04:32:33.359340	0.000062	10.33.	10.33.	TCP	1433	50924	54	1433 → 50924 [ACK] Seq=1689268813 Ack=16458478 Win=256 Len=
80946	04:32:33.359580	0.000240	10.33.	10.33.	TDS	1433	50924	1029	Response
82016	04:32:33.661732	0.302152	10.33.	10.33.	TCP	1433	50924	1029	[TCP Retransmission] 1433 → 50924 [PSH, ACK] Seq=1689268813
82018	04:32:33.666200	0.004468	10.33.	10.33.	TCP	50924	1433	66	[TCP Previous segment not captured] 50924 → 1433 [ACK] Seq=
84171	04:32:34.357081	0.690881	10.33.	10.33.	TCP	50924	1433	99	[TCP Retransmission] 50924 → 1433 [PSH, ACK] Seq=16458478 A
84173	04:32:34.357248	0.000167	10.33.	10.33.	TDS	1433	50924	80	Response
84179	04:32:34.359538	0.002290	10.33.	10.33.	TCP	50924	1433	1514	50924 → 1433 [ACK] Seq=16458523 Ack=1689269814 Win=513 Len=
84311	04:32:34.566797	0.207259	10.33.	10.33.	TCP	1433	50924	54	1433 → 50924 [ACK] Seq=1689269814 Ack=16459983 Win=256 Len=
84313	04:32:34.567322	0.000525	10.33.	10.33.	TCP	50924	1433	860	50924 → 1433 [PSH, ACK] Seq=16459983 Ack=1689269814 Win=513
84315	04:32:34.567938	0.000616	10.33.	10.33.	TDS	1433	50924	1031	Response
84317	04:32:34.568981	0.001043	10.33.	10.33.	TCP	50924	1433	99	50924 → 1433 [PSH, ACK] Seq=16460789 Ack=1689270791 Win=509
84319	04:32:34.569196	0.000215	10.33.	10.33.	TDS	1433	50924	80	Response
84998	04:32:34.878735	0.309539	10.33.	10.33.	TCP	1433	50924	80	[TCP Retransmission] 1433 → 50924 [PSH, ACK] Seq=1689270791
85003	04:32:34.879224	0.000489	10.33.	10.33.	TCP	50924	1433	66	[TCP Previous segment not captured] 50924 → 1433 [ACK] Seq=
85018	04:32:34.887714	0.008490	10.33.	10.33.	TCP	50924	1433	1514	[TCP Retransmission] 50924 → 1433 [ACK] Seq=16460834 Ack=16
85040	04:32:35.097056	0.209342	10.33.	10.33.	TCP	1433	50924	54	1433 → 50924 [ACK] Seq=1689270817 Ack=16462294 Win=256 Len=
85044	04:32:35.097623	0.000567	10.33.	10.33.	TCP	50924	1433	860	[TCP Retransmission] 50924 → 1433 [PSH, ACK] Seq=16462294 A

Frame number 80946 shows a packet leaving the database server, frame 82016 shows the same packet being re-sent by the database server as the application server did not respond to the first packet. The time taken to resend the missing data can be easily calculated just by using the time from frame 80946 through to the time in frame 84171 and is a duration of 0.997 seconds.

Other Points of Note

Workaround

On the evening of Saturday 20th January, the database server was migrated back to DEFESXP124. Analysis of the data collected from each server on Monday 22nd January showed that:

- There was no evidence of packet loss or retransmissions in communications between the two servers.
- Database service response times measured at each server were identical (allowing for a few tens of microseconds of latency).
- The number of concurrent TCP connections between the application and database servers had reduced to 28 due to the improved throughput that resulted from fewer TCP retransmissions.

Conclusions

Our conclusions are as follows:

Ref	Conclusion	Evidence
1	Poor database performance was being caused by resulting in high queue lengths and delays within the FTB application server.	<p>The difference between the database response times measured at the application server and those measured on the database.</p> <p>This is further supported with correlation of the slow response times with periods of TCP Retransmissions.</p>
2	Excessive packet loss occurs somewhere along the path from the FTB application server to the database server is hosted on the VMware ESX server ABCESXP135.	Comparison of the network trace data collected on Friday 19 th January and the data collected on Monday 22 nd January.
3	The excessive packet loss is not related to application load.	The database transaction rate (Database Requests per Second) is consistent throughout the analysis period – see <i>Database Transaction Rate</i> on page 10.

Recommendations

Our recommendations are as follows. For convenience, we have listed them in priority order:

Ref	Issue Addressed	Recommendation	Benefit
1	Poor application performance due to excessive packet loss.	“Pin” the database server to DEFESXP124 to avoid the packet loss problem.	Provides temporary immediate relief while the packet loss problem is being investigated.
2	Poor application performance due to excessive packet loss.	Review the load and error rates on all links that make up the path from the application server to the database server hosted on VMware server ABCESXP135. If the review doesn’t identify the cause of the problem, use matching packet traces captured at intermediate points along the path.	Identifying the cause of the packet loss and fixing it will allow for the free movement of the database server as workload dictates.