Examining a Web-Based Oracle Application

Key Concept—This example shows how AppTransaction Xpert diagnoses and visualizes application and network problems; it is not a step-by-step tutorial. If you have experience with AppTransaction Xpert, you can recreate this study by following the instructions in Recreate the Example. The screen images in this example were captured running AppTransaction Xpert in Windows with the AppTransaction Xpert Decode Module (ADM) installed. If you are working on Linux, or do not have ADM installed, some screens might look different.

In this network, the remote Client accesses an Oracle Application Server to retrieve information from the Oracle Database Server (DB Server). The connection is over a WAN through a T1 circuit.

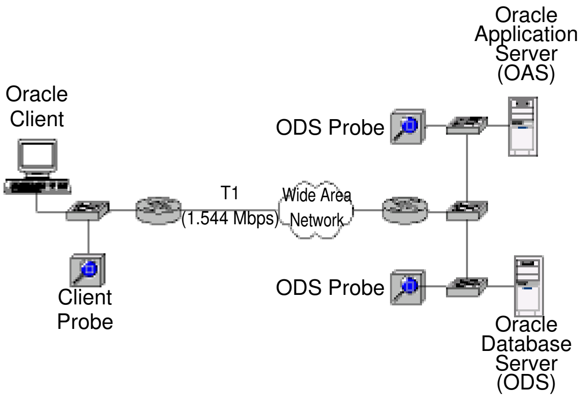
Possible causes for the slow response time fall into two general categories:

- network bottlenecks
- server bottlenecks

This study identifies the exact cause of the performance problems and recommends solutions.

We begin our investigation by identifying and capturing a typical transaction. Probes were placed on the Client, the Application Server, and the DB Server, as shown in the following figure.

Figure 8-1 Network Diagram

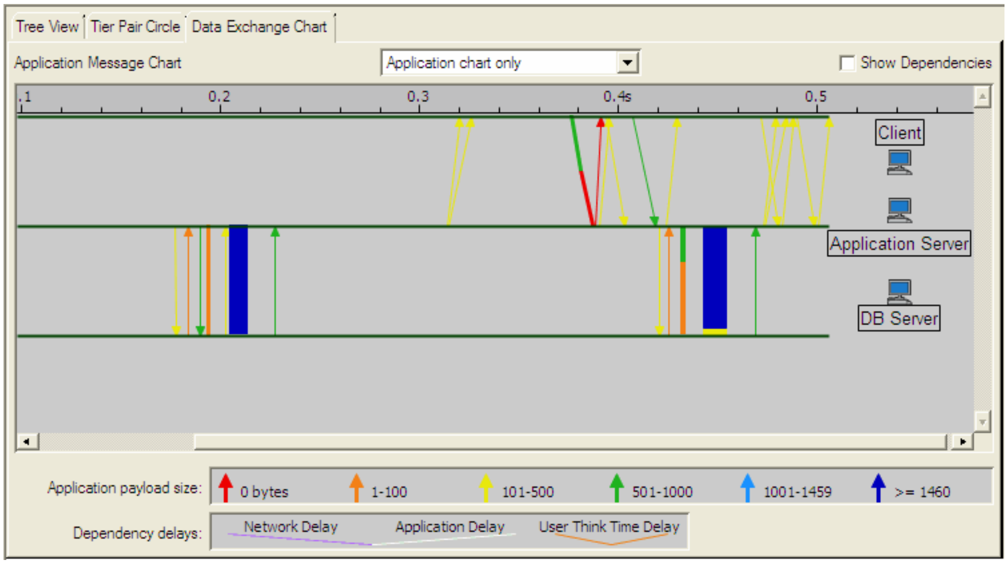


Packet trace captures were taken simultaneously at the probes, then merged and synchronized to get the most accurate measurements of delay at each tier and in the overall network.

Diagnosis

After opening the packet trace, we looked at the transaction in the Data Exchange Chart, which shows the flow of application messages between tiers over time.

Figure 8-2 Data Exchange Chart (Application View)

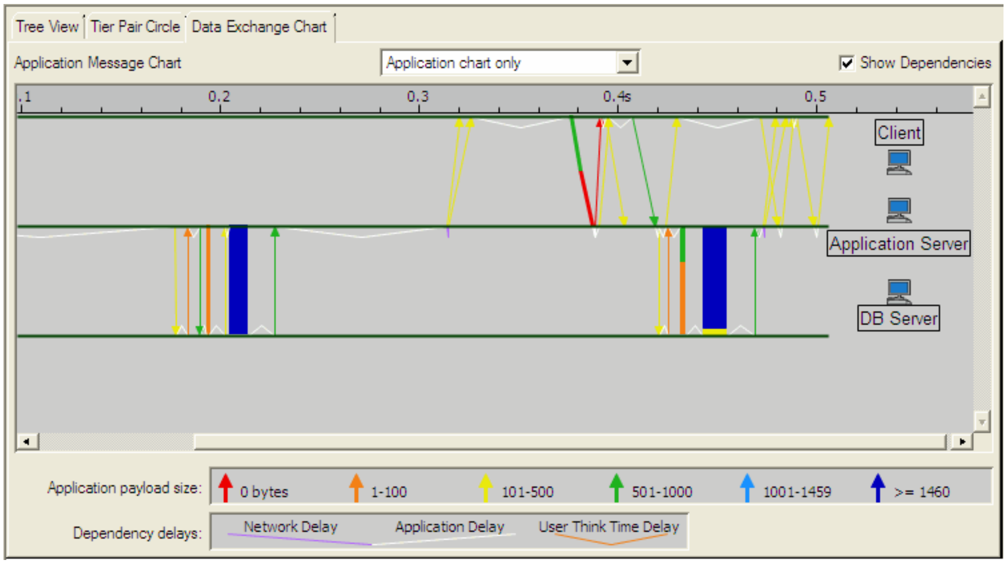


Application Message Chart: Application chart only

□ Show Dependencies

The Data Exchange Chart shows objects of various sizes being transferred between the Client and Application Server and between the Application Server and DB Server. The overall transaction response time is about 0.5 seconds. When the “Show Dependencies” checkbox is selected, the white dependency lines indicate large processing delays on the Application Server and Client tiers.

Figure 8-3 Data Exchange Chart with Dependencies

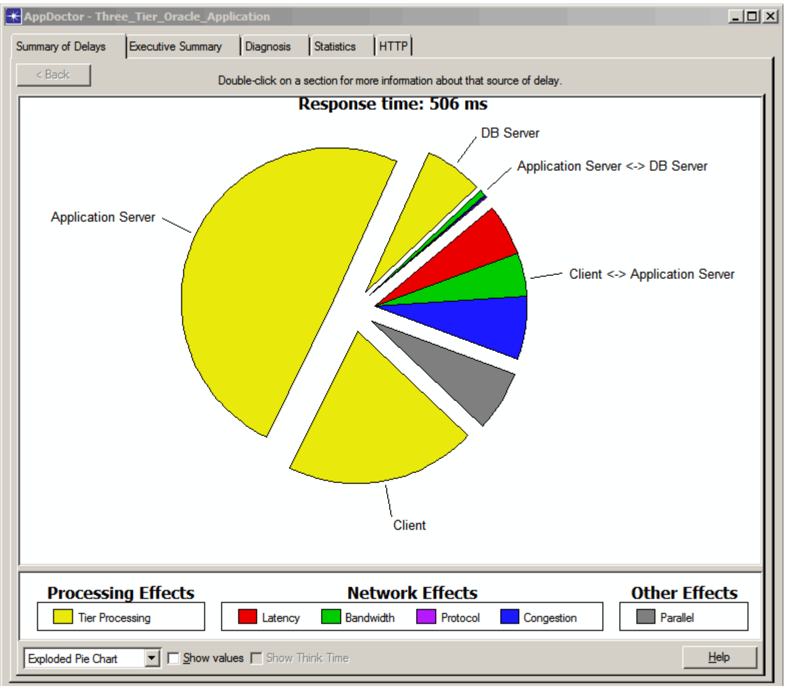


Application Message Chart: Application chart only

☑ Show Dependencies

AppDoctor’s Summary of Delays identifies the factors that contribute to application delay.

Figure 8-4 Summary of Delays for the Oracle Transaction



Double-click on a section for more information about that source of delay.

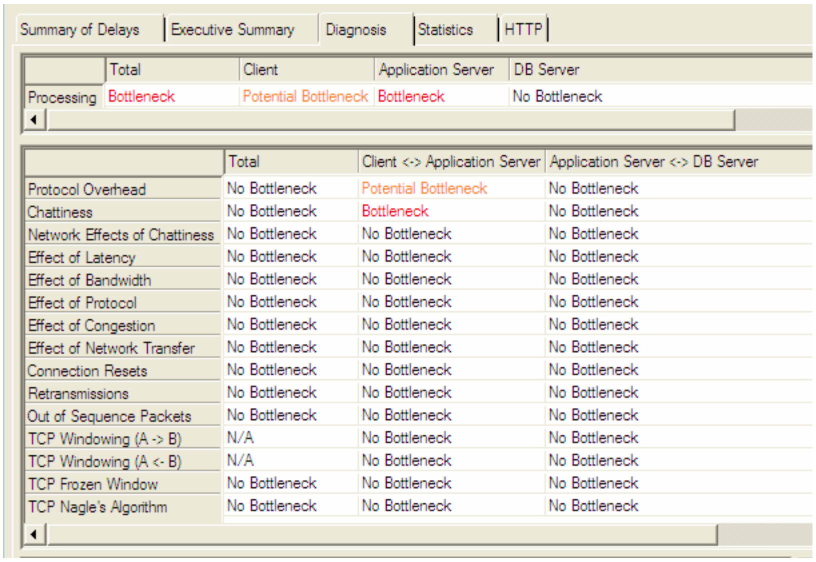
Exploded Pie Chart ☐ Show values ☐ Show Think Time

Processing delay on the Application Server is the largest component of delay for this transaction.

The network-related delays (Latency, Bandwidth, Protocol, and Congestion) are small compared to the processing effects.

AppDoctor’s Diagnosis provides a more granular analysis of possible bottlenecks. This feature tests the current transaction against factors that often cause performance problems in network-based applications. Values that cross a specified threshold are marked as bottlenecks or potential bottlenecks.

Figure 8-5 Diagnosis for the Oracle Transaction



	Total	Client	Application Server	DB Server
Processing	Bottleneck	Potential Bottleneck	Bottleneck	No Bottleneck

	Total	Client <-> Application Server	Application Server <-> DB Server
Protocol Overhead	No Bottleneck	Potential Bottleneck	No Bottleneck
Chattiness	No Bottleneck	Bottleneck	No Bottleneck
Network Effects of Chattiness	No Bottleneck	No Bottleneck	No Bottleneck
Effect of Latency	No Bottleneck	No Bottleneck	No Bottleneck
Effect of Bandwidth	No Bottleneck	No Bottleneck	No Bottleneck
Effect of Protocol	No Bottleneck	No Bottleneck	No Bottleneck
Effect of Congestion	No Bottleneck	No Bottleneck	No Bottleneck
Effect of Network Transfer	No Bottleneck	No Bottleneck	No Bottleneck
Connection Resets	No Bottleneck	No Bottleneck	No Bottleneck
Retransmissions	No Bottleneck	No Bottleneck	No Bottleneck
Out of Sequence Packets	No Bottleneck	No Bottleneck	No Bottleneck
TCP Windowing (A -> B)	N/A	No Bottleneck	No Bottleneck
TCP Windowing (A <- B)	N/A	No Bottleneck	No Bottleneck
TCP Frozen Window	No Bottleneck	No Bottleneck	No Bottleneck
TCP Nagle's Algorithm	No Bottleneck	No Bottleneck	No Bottleneck

Diagnosis confirms that the primary bottleneck is due to Processing Delay at the Application Server. The other potential bottlenecks are:

- Processing at Client
- Protocol Overhead (Client <-> Application Server)
- Chattiness

Processing at Client

Processing delay is due to the file Input/Output, CPU processing, memory access, and so on.

Protocol Overhead

Protocol headers add overhead to each application message. Protocols also send messages that have no application data, such as TCP acknowledgement packets. This overhead can introduce delays by increasing congestion in the network.

Chattiness

The data sent per application turn is small. The numeric value is the number of application bytes per application turn. This may cause significant network delay. Additionally, a significant portion of application processing time can be spent processing requests and responses.

If you have a Chattiness bottleneck without a Network Effects of Chattiness bottleneck, this means that

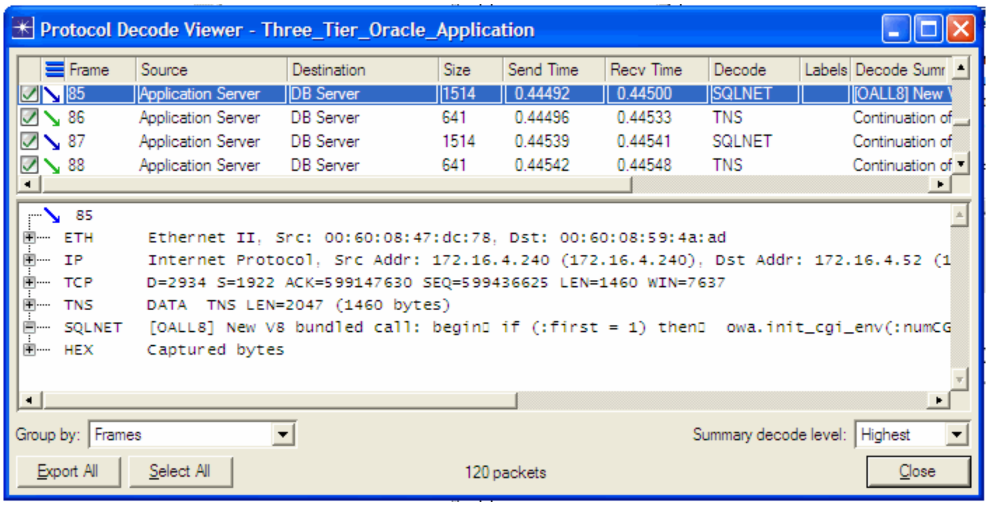
- The application is not incurring significant network delays due to chattiness

- The application may be incurring significant processing delays due to overhead associated with handling many small application requests/responses

- The application’s Network Effects of Chattiness could dramatically increase in a higher-latency network

In addition, the AppTransaction Xpert Decode Module (ADM) enables us to gain more insight into the application.

Figure 8-6 SQL Decode for Frame 85



	Frame	Source	Destination	Size	Send Time	Recv Time	Decode	Labels	Decode Sumr
☑	85	Application Server	DB Server	1514	0.44492	0.44500	SQLNET		[OALL8] New V
☑	86	Application Server	DB Server	641	0.44496	0.44533	TNS		Continuation of
☑	87	Application Server	DB Server	1514	0.44539	0.44541	SQLNET		Continuation of
☑	88	Application Server	DB Server	641	0.44542	0.44548	TNS		Continuation of

```
85
ETH      Ethernet II, Src: 00:60:08:47:dc:78, Dst: 00:60:08:59:4a:ad
IP       Internet Protocol, Src Addr: 172.16.4.240 (172.16.4.240), Dst Addr: 172.16.4.52 (1
TCP      D=2934 S=1922 ACK=599147630 SEQ=599436625 LEN=1460 WIN=7637
TNS      DATA  TNS LEN=2047 (1460 bytes)
SQLNET   [OALL8] New V8 bundled call: begin□ if (:first = 1) then□  owa.init_cgi_env(:numCG
HEX      Captured bytes
```

Group by: Frames

Summary decode level: Highest

120 packets

ADM adds information about each packet that can help us understand the application itself. For example, in Frame 12, we can see an SQL statement.

Conclusion

In this example, the application response time was about half a second. Perhaps not much improvement is necessary. However, using the analysis and diagnostic capabilities of AppTransaction Xpert, we can highlight areas that can be improved, supplying helpful information to the application developers.

AppTransaction Xpert with AppTransaction Xpert Decode Module lets you troubleshoot application performance and identify the potential causes.

Recreate the Example

Load the Transaction Analyzer model “Three_Tier_Oracle_Application” (from <*reldir*>\sys\examples\AppTransaction Xpert\examples) or perform the following procedure.

Procedure 8-1 Recreating the Example

1 Open the following packet traces in AppTransaction Xpert:
(**File** > **Open Packet Trace(s)** >
In Transaction Analyzer (Simultaneous Captures)…)

- Three_Tier_Oracle_Remote_Client.enc
- Three_Tier_Oracle_Application_Server.enc
- Three_Tier_Oracle_Database_Server.enc

2 Rename the tiers:

- Rename 192.168.50.170 to Client
- Rename 172.16.4.240 to Application Server
- Rename 172.16.4.52 to DB Server

3 Set bandwidth and latency between tiers:
(**AppDoctor** > **Refine Network Effects**…)

- Set the remote bandwidth from “Site 1” to “Site 2” to T1 (1.544 Mbps)
- Set the remote bandwidth from “Site 2” to “Site 3” to 100,000 (100 Mbps)
- Latency between "Site 1" and "Site 3" should be detected as approximately 1.7 ms

End of Procedure 8-1