FTP Application

This tutorial shows how you can use AppTransaction Xpert to

- Identify the true causes of poor response time
- Document the problems that are found

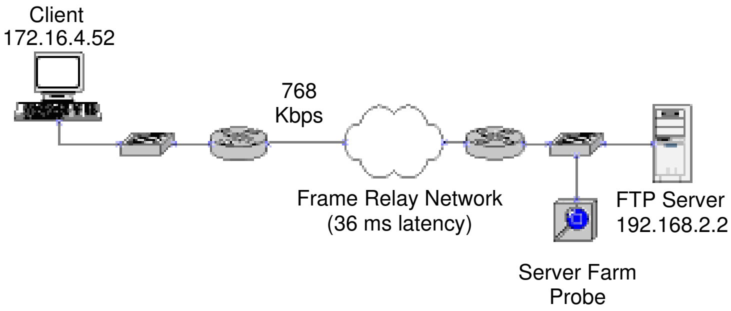
Key Concept—The screen images in this tutorial were captured while running AppTransaction Xpert in Windows with both AppTransaction Xpert Decode Module (ADM) and Wireshark external decoder. If you are working on Linux or if you do not have ADM and/or Wireshark, some screens might look different.

In this tutorial, you will

- Open a packet trace
- Examine the application in the Data Exchange Chart and Treeview
- Examine the protocol decode for a frame
- Examine AppDoctor’s Summary of Delays, Diagnosis, and Statistics
- Create graphs of statistics
- Publish results to a web report

The subject application runs on the network topology shown in the following figure. The client connects to the server over a 768 Kbps Frame Relay circuit with 36 ms of latency.

Figure 2-1 Example Network Topology



The subject application (FTP) downloads a 1 MB file in 37 seconds. Normally, the download time for a file this size should be about 11 seconds.

Open the Packet Trace

When using AppTransaction Xpert to solve an application problem, you must accurately characterize the network where the packet trace was captured. The primary information required is the bandwidth and latency between the capture probe and the relevant tier machines. This information allows AppTransaction Xpert to differentiate between delays caused by the network and the servers.

In this case, the packet trace is a single file captured at the FTP server.

Procedure 2-1 Opening a Packet Trace

1 Choose **File > Open Packet Trace(s) > In Transaction Analyzer (Single Capture)…**

➥The “Open a Capture File” dialog box appears.

2 Add the **FTP_with_loss_server.enc** file:

2.1 Navigate to ***<reldir>*\sys\examples\AppTransaction Xpert\examples** and select **FTP_with_loss_server.enc**.

<reldir> is the release directory where AppTransaction Xpert is installed. In the Windows environment, this is typically C:\Program Files\OPNET\<*release number*>.

2.2 Select the file and click **Open**.

➥The packet trace opens in the Transaction Analyzer window.

If the **Getting Started Analyzing an Application Transaction** window appears, click Close.

End of Procedure 2-1

Visualize the Application

After opening the packet trace, AppTransaction Xpert shows the Tree View, with information about the TCP connections. We'll come back to the Tree View. But to start, let's go to the Data Exchange Chart (DEC), which depicts the flow of application traffic between tiers.

Procedure 2-2 Viewing the Application

1 Click the Data Exchange Chart tab.

Alternately, click the Data Exchange Chart tool button.

2 If the DEC on your screen does not show 192.168.2.2 (FTP Server) as the top tier, drag the tier label from the bottom to the top.

3 For easy identification, rename the tiers:

3.1 Right-click on the 172.16.4.52 tier label and select **Rename 172.16.4.52**. Then from the submenu, select **Edit…**

3.2 In the **Rename Tier** dialog box, enter **Client** and click **OK**.

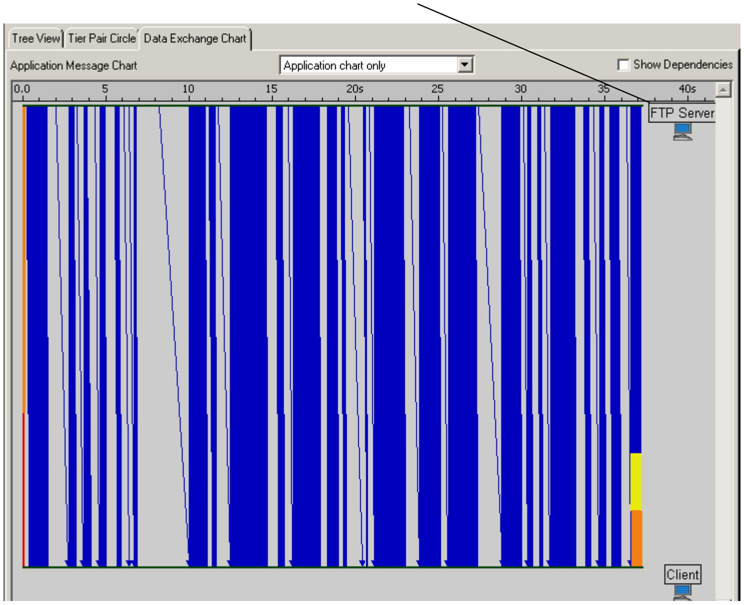
3.3 Right-click on the 192.168.2.2 tier label and select **Rename 192.168.2.2**. Then from the submenu, select **Edit…**

3.4 In the **Rename Tier** dialog box, enter **FTP Server** and click **OK**.

Your screen should match the following figure.

Figure 2-2 Data Exchange Chart (Application View)

FTP Server is the top tier



The Data Exchange Chart shows the data transferred between tiers on a time line.

The DEC for this FTP download shows that most of the communication between the FTP Server and the Client is composed of messages with application payloads of 1,460 bytes or higher (dark blue messages).

The Data Exchange Chart can display three different views.

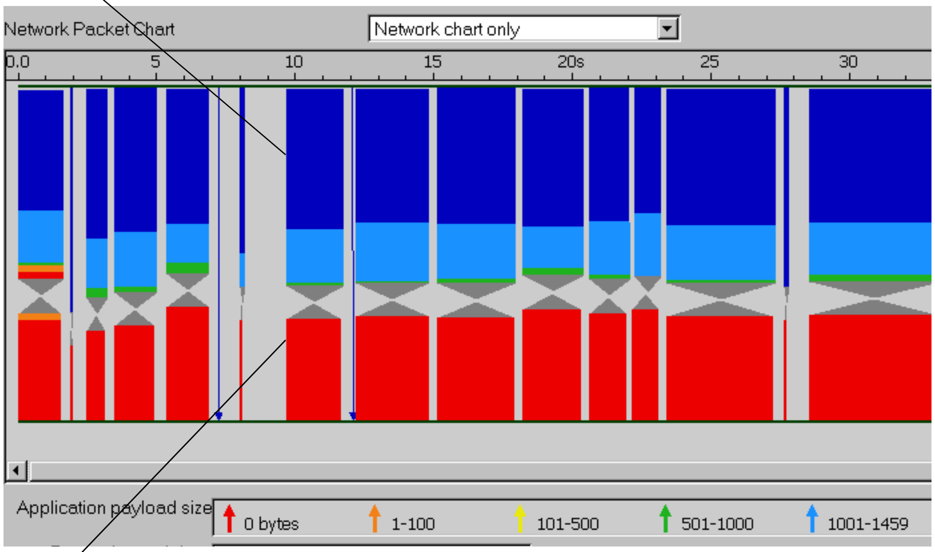
- The *Application Chart Only* view is the default. The Application Message Chart shows the flow of application traffic between tiers.
- The *Network Chart Only* view shows the flow of network traffic between tiers, including the effects of network protocols on application traffic. Network protocols split packets into segments, add headers, and often include mechanisms to ensure reliable data transfer. These network protocol effects can influence application behavior.
- The application and network charts view is a combination of the two views described above.

4 Choose **Network Chart Only** from the drop-down menu in the middle of the dialog box.

5 Differentiate the messages flowing in different directions by choosing **View** > **Split Groups**.

Figure 2-3 Data Exchange Chart (Network View)

FTP Server to Client messages: mainly large messages



Client to FTP Server messages: many 0-bytes messages

6 Now the messages are split into two groups and you can see more details:

- The upper-group represents messages from the FTP Server to the Client and the lower-group represents messages from the Client to the FTP Server.
- Mainly large messages (light blue and dark blue) flow from the FTP Server to the Client.

- Zero Byte messages (red) flow from the Client to the FTP Server. Because the red traffic appears only in the Network View, you know that these messages represent some of the protocol effects on this application.

For a better understanding of the traffic, you will zoom in on a transaction. To understand how the Application Chart and Network Chart views differ, you will view both simultaneously.

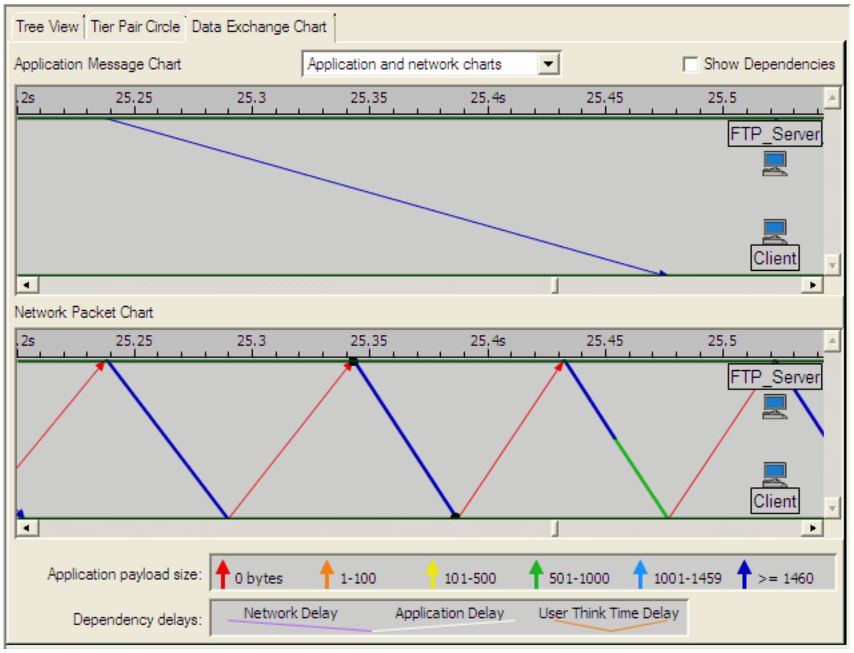
7 Choose **Application and Network Charts** from the drop-down menu in the middle of the dialog box.

8 To disable the split groups view, choose **View** > **Split Groups**.

9 Choose **View** > **Set Visible Time Range**.

10 In the **Specify Time Range** dialog box, set **Start time** to **25.2** and **End time** to **25.5**, then click **OK**.

Figure 2-4 Zoomed View of the Application and Network Charts



11 Notice the differences between the two views:

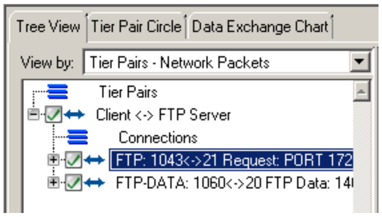
- The Application Chart shows a single 8192-byte message flowing from the FTP Server to the Client. To show the size, place the cursor on the message to show the tooltip. The payload is **8192**.
- The Network Chart shows that this application message causes many packets to flow over the network. These packets are a mix of large (blue and green) packets from the FTP Server to the Client and small (red) packets from the Client to the FTP Server. As the red color indicates, these packets contain 0 bytes of application data. They are the acknowledgements sent by TCP.

Further information about the TCP connections is available in the Tree View. The Tree View provides a text listing and analysis of all of the packets in the original packet trace. Values such as total application Bytes, network traffic, packet send times, and packet size are available. In some cases the Tree View separates traffic into individual transactions.

12 Click on the Tree View tool button.

13 Expand the Tier Pairs tree.

Figure 2-5 Tier Pair Expanded



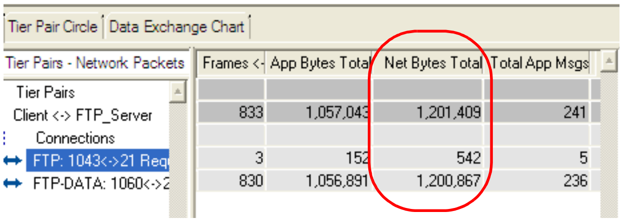
Tree View | Tier Pair Circle | Data Exchange Chart

View by: Tier Pairs - Network Packets

- Tier Pairs
  - ☑ Client <-> FTP Server
    - Connections
      - ☑ FTP: 1043<->21 Request: PORT 172
      - ☑ FTP-DATA: 1060<->20 FTP Data: 14

14 Scroll to the right in the middle pane to display the “Net Bytes Total” column.

Figure 2-6 Middle Pane (Partial View of Tree View Dialog Box)



Tier Pair Circle | Data Exchange Chart

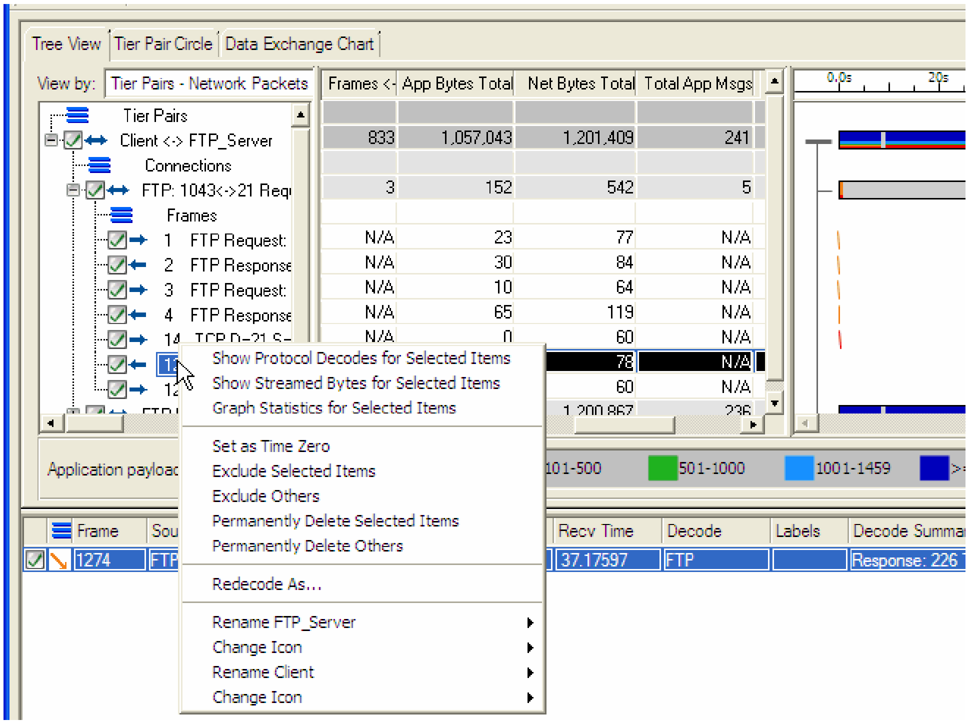
Tier Pairs - Network Packets	Frames <-	App Bytes Total	Net Bytes Total	Total App Msgs
Tier Pairs				
Client <-> FTP_Server	833	1,057,043	1,201,409	241
Connections				
FTP: 1043<->21 Req	3	152	542	5
FTP-DATA: 1060<->2	830	1,056,891	1,200,867	236

Notice that this application has two distinct TCP connections: one carries 542 bytes; the second carries about 1.2 MB. This behavior is typical of FTP. One connection sends control information (FTP commands); separate connections transfer the actual data files.

15 In the Tier Pairs tree, expand the first connection.

16 Display the decode for Frame 1274 by right-clicking on that transaction and selecting **Show Protocol Decodes For Selected Items**.

Figure 2-7 First TCP Connection Expanded and Pop-up Menu Displayed


Tree View
Tier Pair Circle
Data Exchange Chart
View by:
Tier Pairs - Network Packets
0.0s
20s
Show Protocol Decodes for Selected Items
Show Streamed Bytes for Selected Items
Graph Statistics for Selected Items
Set as Time Zero
Exclude Selected Items
Exclude Others
Permanently Delete Selected Items
Permanently Delete Others
Redecode As...
Rename FTP_Server
Change Icon
Rename Client
Change Icon

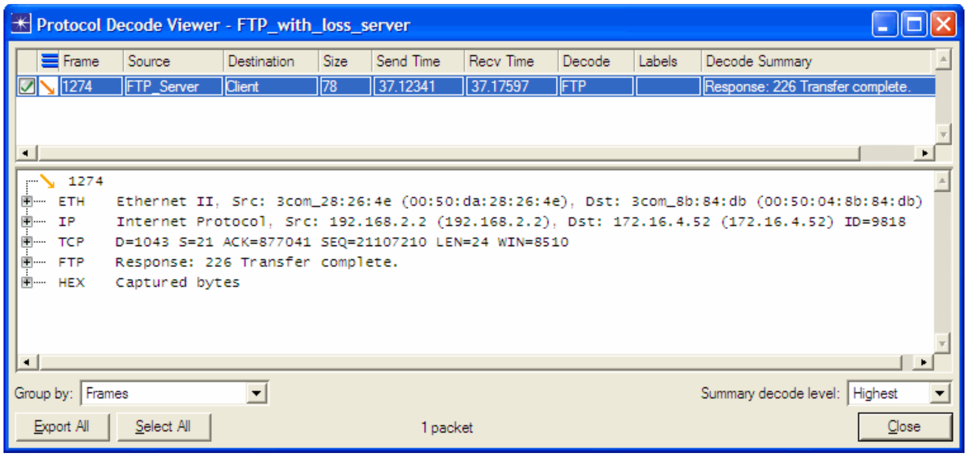
	Frames <-	App Bytes Total	Net Bytes Total	Total App Msgs
Tier Pairs				
Client <-> FTP_Server	833	1,057,043	1,201,409	241
Connections				
FTP: 1043<->21 Req	3	152	542	5
Frames				
1 FTP Request:	N/A	23	77	N/A
2 FTP Response	N/A	30	84	N/A
3 FTP Request:	N/A	10	64	N/A
4 FTP Response	N/A	65	119	N/A
14 TCP D=21 S=	N/A	0	60	N/A
12			78	N/A
1			60	N/A
FTP			1,200,867	236

Application payload ... 101-500 🟩 501-1000 🟦 1001-1459 🟦 >

Frame	Sou	Recv Time	Decode	Labels	Decode Summa
☑ 1274	FTP	37.17597	FTP		Response: 226

➥ The Protocol Decode Viewer appears. If you have a license for the AppTransaction Xpert Decode Module (ADM), the decodes will look as follows.

Figure 2-8 Protocol Decode for Frame 1274 (ADM Installed)



Frame	Source	Destination	Size	Send Time	Recv Time	Decode	Labels	Decode Summary
☑ 1274	FTP_Server	Client	78	37.12341	37.17597	FTP		Response: 226 Transfer complete.

```
1274
ETH   Ethernet II, Src: 3com_28:26:4e (00:50:da:28:26:4e), Dst: 3com_8b:84:db (00:50:04:8b:84:db)
IP    Internet Protocol, Src: 192.168.2.2 (192.168.2.2), Dst: 172.16.4.52 (172.16.4.52) ID=9818
TCP   D=1043 S=21 ACK=877041 SEQ=21107210 LEN=24 WIN=8510
FTP   Response: 226 Transfer complete.
HEX   Captured bytes
```

Group by: Frames

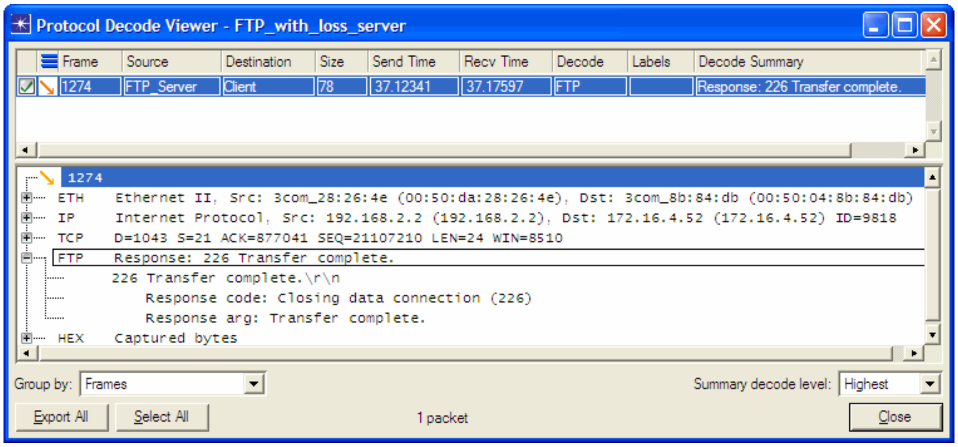
Summary decode level: Highest

1 packet

17 Notice that the message contains FTP data.

18 If you have Wireshark installed, you can expand the FTP data to see the decoded FTP “transfer complete” message:

Figure 2-9 Expanded Protocol Decode for Frame 1274 (ADM & Wireshark Installed)



Protocol Decode Viewer - FTP_with_loss_server

Frame	Source	Destination	Size	Send Time	Recv Time	Decode	Labels	Decode Summary
☑ 1274	FTP_Server	Client	78	37.12341	37.17597	FTP		Response: 226 Transfer complete.

```
1274
ETH    Ethernet II, Src: 3com_28:26:4e (00:50:da:28:26:4e), Dst: 3com_8b:84:db (00:50:04:8b:84:db)
IP     Internet Protocol, Src: 192.168.2.2 (192.168.2.2), Dst: 172.16.4.52 (172.16.4.52) ID=9818
TCP    D=1043 S=21 ACK=877041 SEQ=21107210 LEN=24 WIN=8510
FTP    Response: 226 Transfer complete.
       226 Transfer complete.\r\n
           Response code: Closing data connection (226)
           Response arg: Transfer complete.
HEX    Captured bytes
```

Group by: Frames

Summary decode level: Highest

1 packet

19 Close the Protocol Decode Viewer.

End of Procedure 2-2

Analyze with AppDoctor

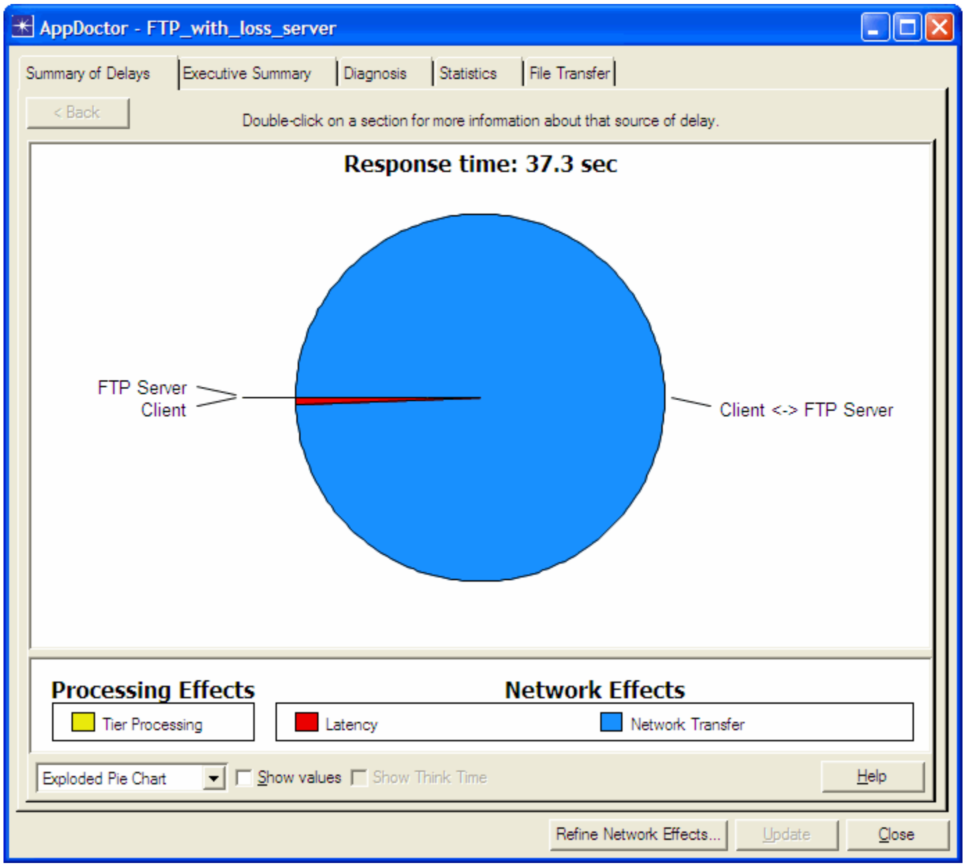
Now that you have become familiar with the traffic between the FTP Server and the Client, you can use AppDoctor to identify the reason for the slow download.

AppDoctor's Summary of Delays provides insight into the root cause of the overall Tier Processing delay. However, before looking at the AppDoctor results, we must refine the network effects (bandwidth and latency) for an accurate analysis.

Procedure 2-3 Analyzing with AppDoctor

1 Choose **AppDoctor** > **Summary of Delays**.

➥ The AppDoctor Summary of Delays window appears.

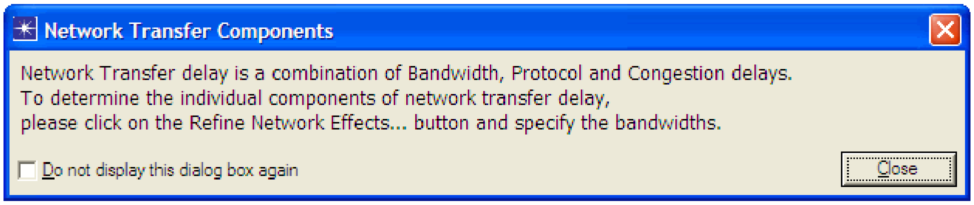


Double-click on a section for more information about that source of delay.

Note that the analysis shows that the majority of the delay is due to Network Transfer.

2 Double-click on the **Network Transfer** (blue) area of the pie for more information about this source of delay.

➥ The following message box appears.



Network Transfer Components

Network Transfer delay is a combination of Bandwidth, Protocol and Congestion delays.
To determine the individual components of network transfer delay,
please click on the Refine Network Effects... button and specify the bandwidths.

□ Do not display this dialog box again

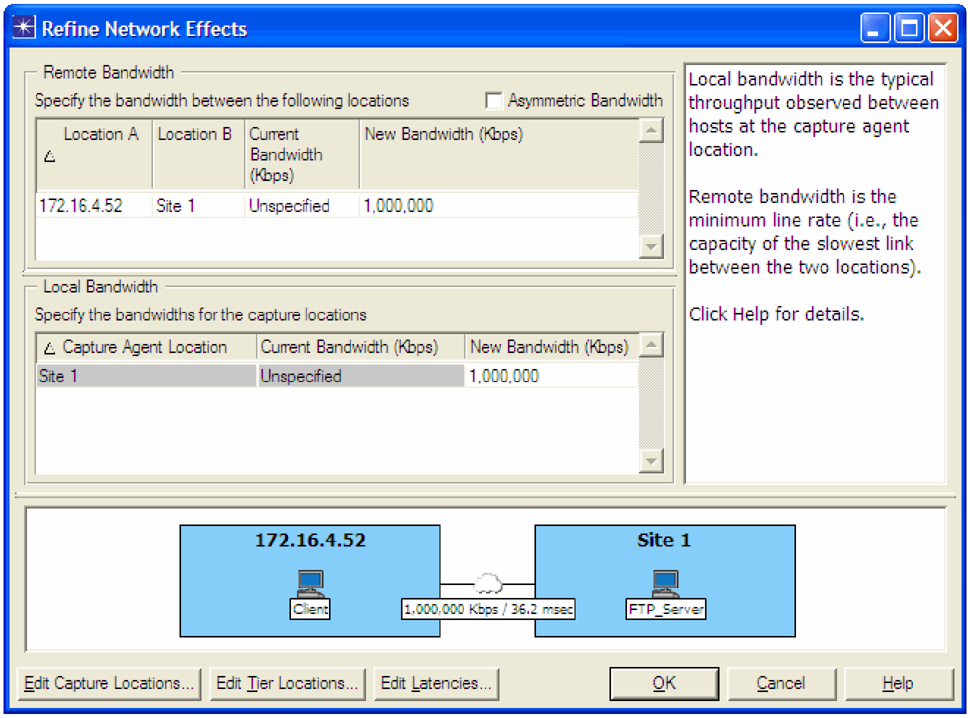
This message indicates that the network effects must be defined to accurately analyze the delay.

3 Click **Close** to close the message box and click **< Back** to return to the Summary of Delays pie graph.

4 Define the network effects:

4.1 Click the **Refine Network Effects…** button.

➥ The Refine Network Effects dialog box appears.



Refine Network Effects

Remote Bandwidth

Specify the bandwidth between the following locations □ Asymmetric Bandwidth

Location A	Location B	Current Bandwidth (Kbps)	New Bandwidth (Kbps)
172.16.4.52	Site 1	Unspecified	1,000,000

Local Bandwidth

Specify the bandwidths for the capture locations

Capture Agent Location	Current Bandwidth (Kbps)	New Bandwidth (Kbps)
Site 1	Unspecified	1,000,000

Local bandwidth is the typical throughput observed between hosts at the capture agent location.

Remote bandwidth is the minimum line rate (i.e., the capacity of the slowest link between the two locations).

Click Help for details.

4.2 Specify the bandwidth for the Server Farm network segment.

For the “Local Bandwidth”, enter **100,000** (100 Mbps) in the **New Bandwidth (Kbps)** field.

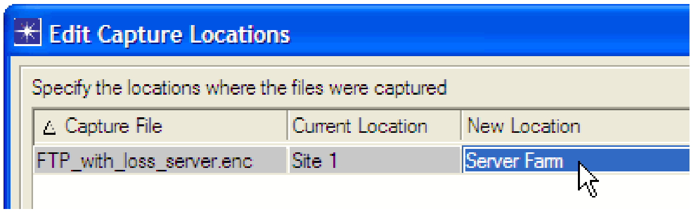
4.3 Set the bandwidth between Server Farm and Client.

For the “Remote Bandwidth”, enter **768** in the **New Bandwidth (Kbps)** field.

4.4 Click the **Edit Capture Locations…** button.

➥ The Edit Capture Locations dialog box appears.

4.5 Enter **Server Farm** in the **New Location** field.



Edit Capture Locations

Specify the locations where the files were captured

Capture File	Current Location	New Location
FTP_with_loss_server.enc	Site 1	Server Farm

4.6 Click **OK** to close the Edit Capture Locations dialog box.

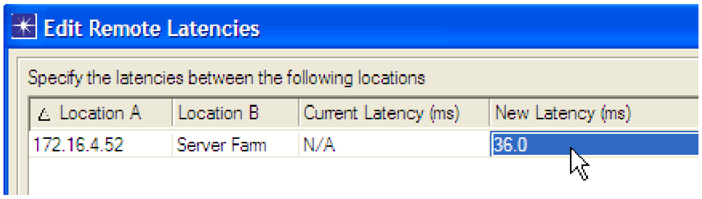
Note—For this tutorial, it is not necessary to edit the tier locations. However, if you click the **Edit Tier Locations…** button, you can enter information about the location of tiers with respect to a capture location. This stage is important if any relevant tiers are not located on the same local segment as the original capture probes.

4.7 Click the **Edit Latencies…** button.

➥ The **Edit Remote Latencies** dialog box appears.

4.8 Set the one-way latency between the Client and the Server Farm.

Enter **36** in the **New Latency (ms)** field.



Edit Remote Latencies

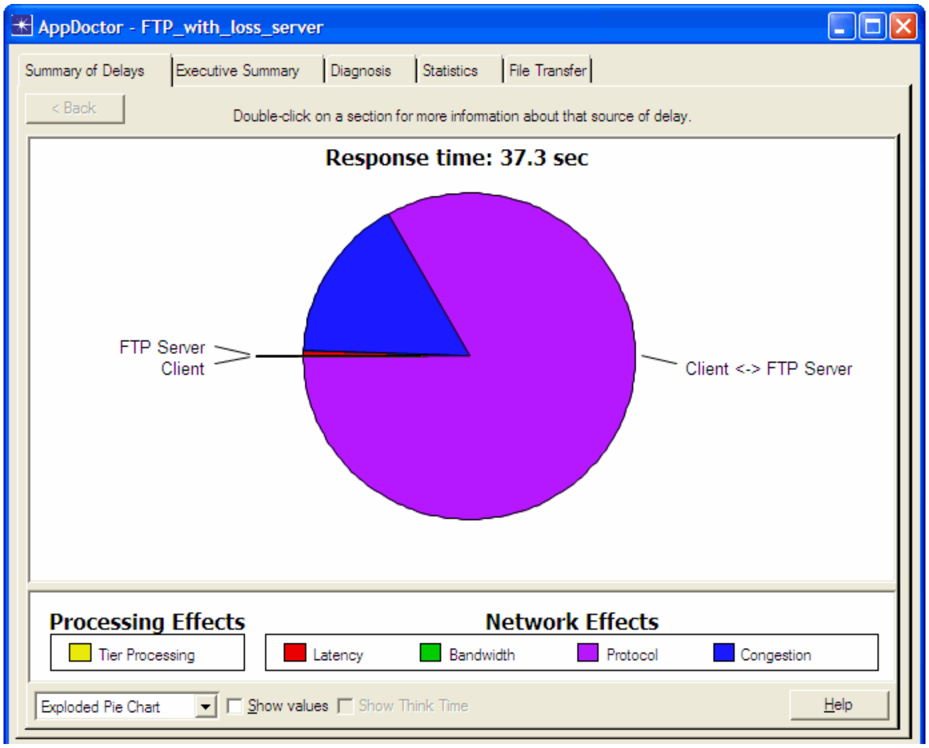
Specify the latencies between the following locations

Location A	Location B	Current Latency (ms)	New Latency (ms)
172.16.4.52	Server Farm	N/A	36.0

4.9 Click **OK** to close the Edit Remote Latencies dialog box.

4.10 Click **OK** to close the Refine Network Effects dialog box.

5 Notice that the AppDoctor Summary of Delays is updated based on the specified network effects. Also notice that the results no longer report Network Transfer as a delay.



6 Place your mouse cursor in the purple area of the pie chart.

➥ The tooltip for Protocol appears. Notice that the largest contributing factor to the application response time is Protocol Effects.

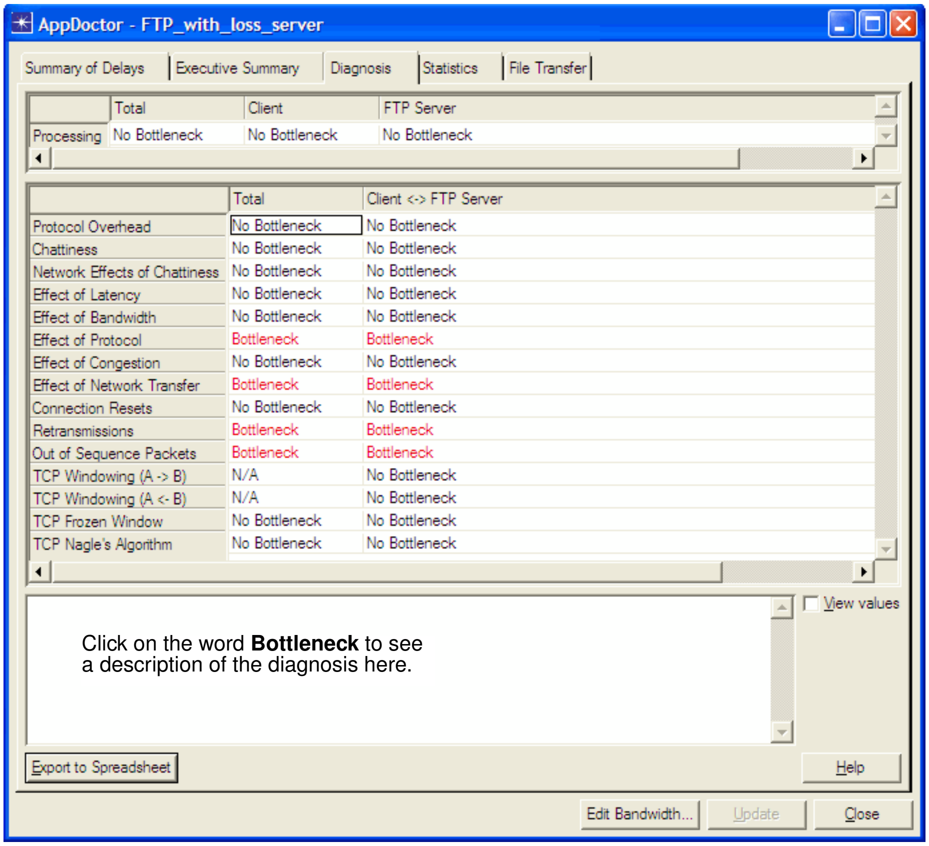
7 Notice that Tier Processing delay (server processing) by both the Client and the FTP Server is not a contributing factor to the application response time.

You conclude that a server upgrade will have little or no effect on download time. A bandwidth upgrade will decrease the download time by no more than about 30 percent.

The Diagnosis function of AppDoctor will provide further insight into the cause of the Protocol delay.

8 Click the **Diagnosis** tab.

Figure 2-10 Diagnosis for the File Transfer



	Total	Client	FTP Server
Processing	No Bottleneck	No Bottleneck	No Bottleneck

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

Click on the word **Bottleneck** to see
a description of the diagnosis here.

9 The diagnosis reveals four bottlenecks:

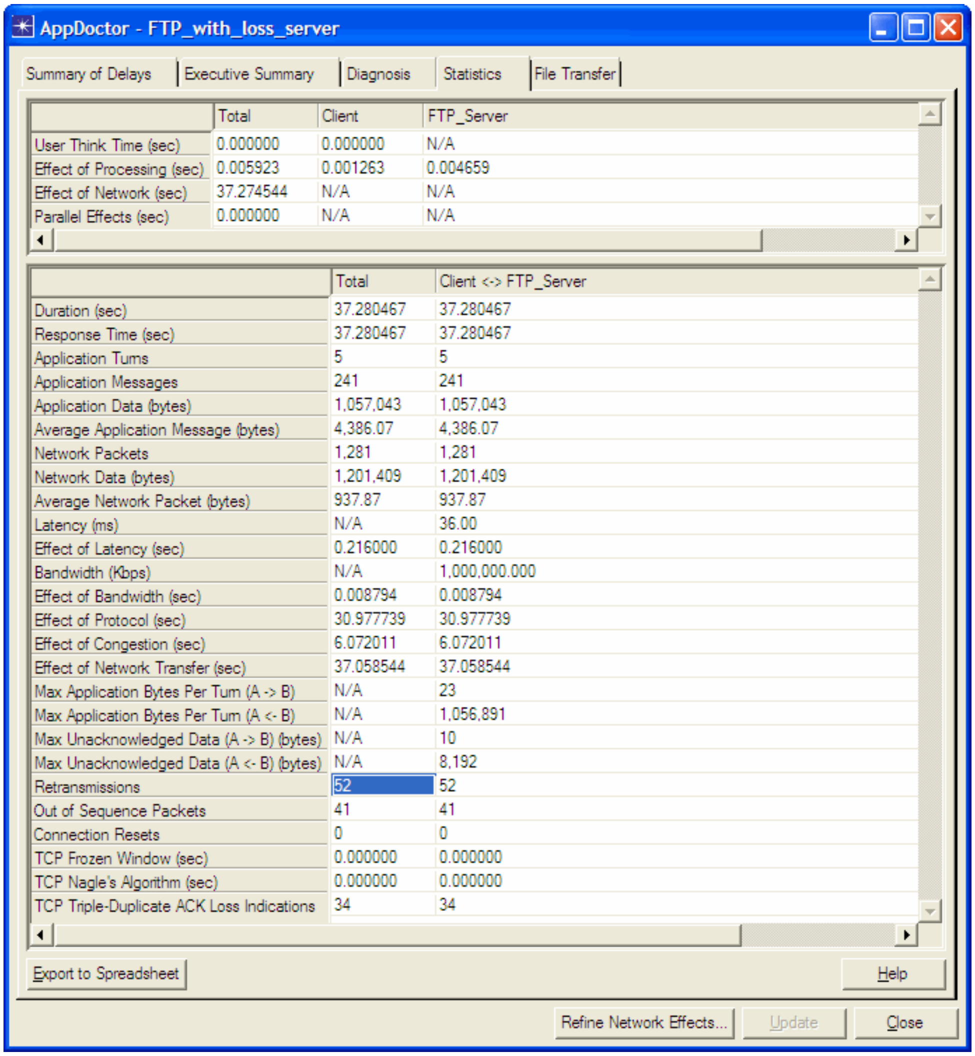
- Effect of Protocol
- Effect of Network Transfer (Bandwidth)
- Retransmissions
- Out of Sequence Packets

The Summary of Delays pie chart showed Bandwidth Delay and Protocol Delay as bottlenecks. One factor that contributes to Protocol Delay is retransmissions. So it is no surprise that here, in the more detailed Diagnosis, you see Retransmissions listed as a bottleneck. The Out of Sequence Packets, also listed as a bottleneck, are a side-effect of the retransmissions. You will explore the retransmissions issue further. Correcting that issue will probably also cure the Out of Sequence Packets problem.

AppDoctor also provides summary statistics for the application transaction. You will examine two in particular: retransmissions and network packets.

10 Click the **Statistics** tab.

Figure 2-11 AppDoctor Statistics



	Total	Client	FTP_Server
User Think Time (sec)	0.000000	0.000000	N/A
Effect of Processing (sec)	0.005923	0.001263	0.004659
Effect of Network (sec)	37.274544	N/A	N/A
Parallel Effects (sec)	0.000000	N/A	N/A

	Total	Client <-> FTP_Server
Duration (sec)	37.280467	37.280467
Response Time (sec)	37.280467	37.280467
Application Turns	5	5
Application Messages	241	241
Application Data (bytes)	1,057,043	1,057,043
Average Application Message (bytes)	4,386.07	4,386.07
Network Packets	1,281	1,281
Network Data (bytes)	1,201,409	1,201,409
Average Network Packet (bytes)	937.87	937.87
Latency (ms)	N/A	36.00
Effect of Latency (sec)	0.216000	0.216000
Bandwidth (Kbps)	N/A	1,000,000.000
Effect of Bandwidth (sec)	0.008794	0.008794
Effect of Protocol (sec)	30.977739	30.977739
Effect of Congestion (sec)	6.072011	6.072011
Effect of Network Transfer (sec)	37.058544	37.058544
Max Application Bytes Per Turn (A -> B)	N/A	23
Max Application Bytes Per Turn (A <- B)	N/A	1,056,891
Max Unacknowledged Data (A -> B) (bytes)	N/A	10
Max Unacknowledged Data (A <- B) (bytes)	N/A	8,192
Retransmissions	52	52
Out of Sequence Packets	41	41
Connection Resets	0	0
TCP Frozen Window (sec)	0.000000	0.000000
TCP Nagle's Algorithm (sec)	0.000000	0.000000
TCP Triple-Duplicate ACK Loss Indications	34	34

11 Notice that 52 retransmissions occurred during a file transfer composed of 1,281 packets, yielding a retransmission rate of 4 percent. This is a fairly high rate and is the primary cause of the protocol delay.

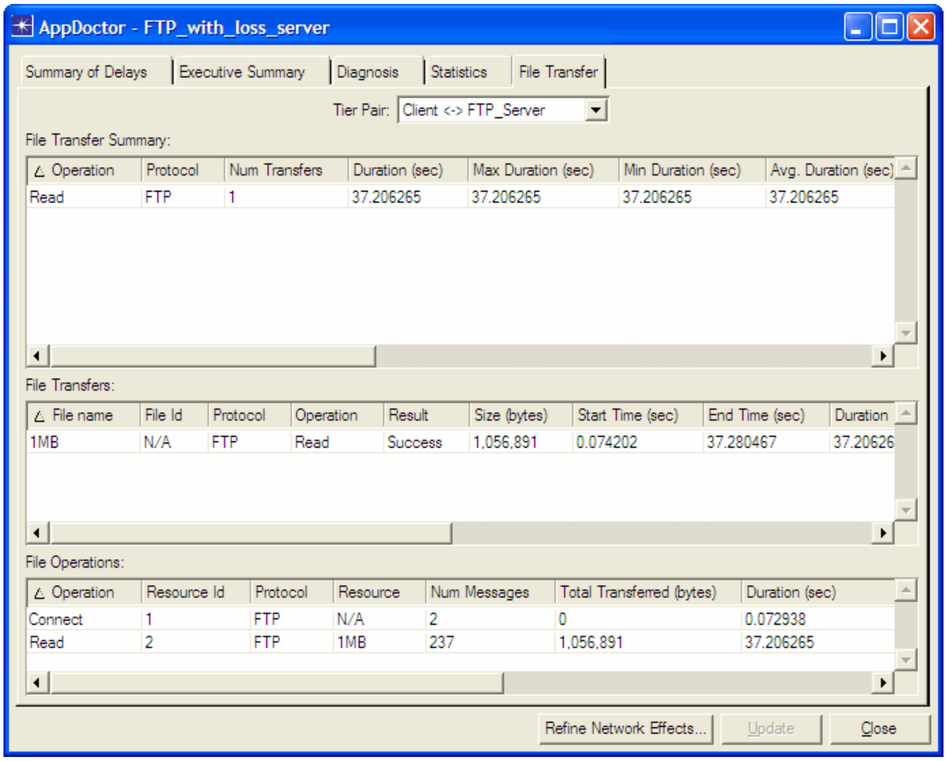
The protocol delay occurs because of two aspects of TCP behavior. Both occur because TCP is responsible for network reliability.

- First, to detect packet loss, TCP uses timers for each packet. When the network drops a packet, TCP waits until the timer expires, then sends the retransmission. Waiting for the timers to expire can greatly increase the application response time.
- Second, packet loss reduces the number of simultaneous packets that TCP sends. This congestion control reduces the throughput of the application and increases the response time.

12 Click the **File Transfer** tab.

This page shows summary information and statistics for the file transfers.

Figure 2-12 File Transfer Statistics



Tier Pair: Client <-> FTP_Server

File Transfer Summary:

Operation	Protocol	Num Transfers	Duration (sec)	Max Duration (sec)	Min Duration (sec)	Avg. Duration (sec)
Read	FTP	1	37.206265	37.206265	37.206265	37.206265

File Transfers:

File name	File Id	Protocol	Operation	Result	Size (bytes)	Start Time (sec)	End Time (sec)	Duration
1MB	N/A	FTP	Read	Success	1,056,891	0.074202	37.280467	37.20626

File Operations:

Operation	Resource Id	Protocol	Resource	Num Messages	Total Transferred (bytes)	Duration (sec)
Connect	1	FTP	N/A	2	0	0.072938
Read	2	FTP	1MB	237	1,056,891	37.206265

13 Close the AppDoctor window.

End of Procedure 2-3

Examine Statistics

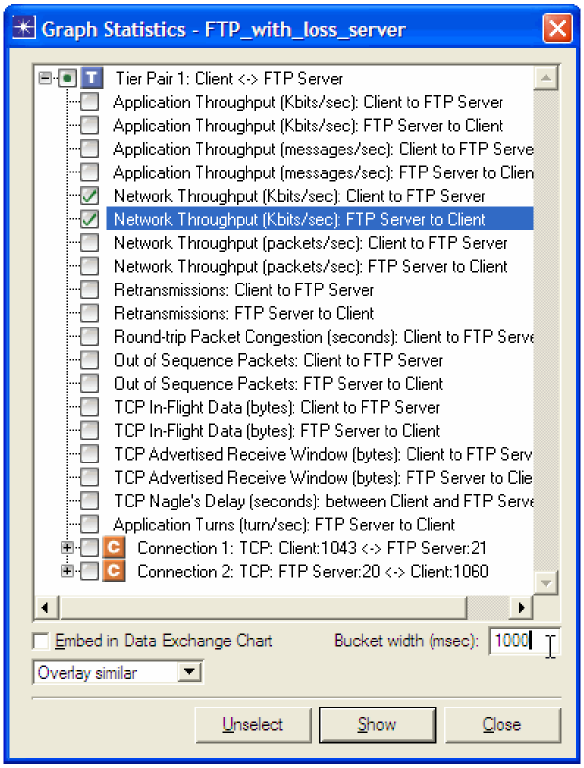
To view the actual network throughput, use the Graph Statistics feature.

Procedure 2-4 Examining Statistics

1 Choose **View > Graph Statistics**.

➥ The Graph Statistics dialog box appears.

2 Select the two Network Throughput statistics that measure kbits/sec.



Graph Statistics - FTP_with_loss_server

- ⊟ ● T Tier Pair 1: Client <-> FTP Server
  - □ Application Throughput (Kbits/sec): Client to FTP Server
  - □ Application Throughput (Kbits/sec): FTP Server to Client
  - □ Application Throughput (messages/sec): Client to FTP Serve
  - □ Application Throughput (messages/sec): FTP Server to Clien
  - ☑ Network Throughput (Kbits/sec): Client to FTP Server
  - ☑ Network Throughput (Kbits/sec): FTP Server to Client
  - □ Network Throughput (packets/sec): Client to FTP Server
  - □ Network Throughput (packets/sec): FTP Server to Client
  - □ Retransmissions: Client to FTP Server
  - □ Retransmissions: FTP Server to Client
  - □ Round-trip Packet Congestion (seconds): Client to FTP Serve
  - □ Out of Sequence Packets: Client to FTP Server
  - □ Out of Sequence Packets: FTP Server to Client
  - □ TCP In-Flight Data (bytes): Client to FTP Server
  - □ TCP In-Flight Data (bytes): FTP Server to Client
  - □ TCP Advertised Receive Window (bytes): Client to FTP Serv
  - □ TCP Advertised Receive Window (bytes): FTP Server to Clie
  - □ TCP Nagle's Delay (seconds): between Client and FTP Serve
  - □ Application Turns (turn/sec): FTP Server to Client
  - ⊞ □ C Connection 1: TCP: Client:1043 <-> FTP Server:21
  - ⊞ □ C Connection 2: TCP: FTP Server:20 <-> Client:1060

□ Embed in Data Exchange Chart

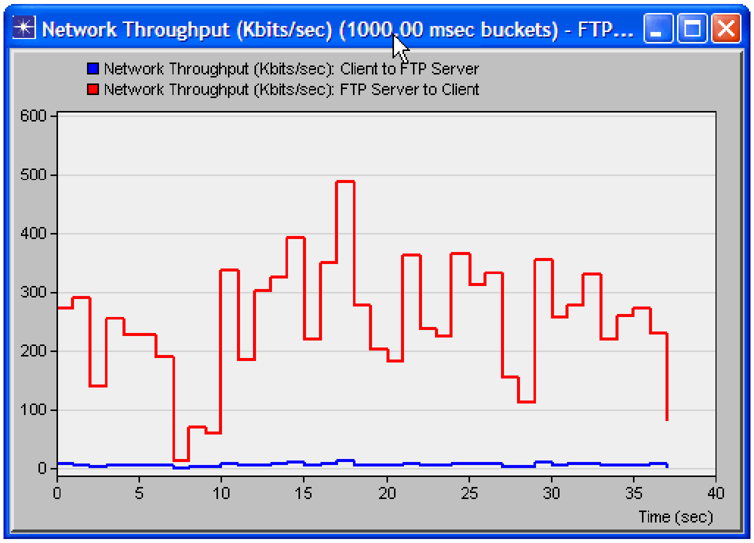
Bucket width (msec): 1000

Overlay similar

3 Set the **Bucket width (msec)** to **1000**.

4 Click **Show**.

➥ The statistics graph for network throughput appears.


Network Throughput (Kbits/sec) (1000.00 msec buckets) - FTP...
🟦 Network Throughput (Kbits/sec): Client to FTP Server
🟥 Network Throughput (Kbits/sec): FTP Server to Client
600
500
400
300
200
100
0
0
5
10
15
20
25
30
35
40
Time (sec)

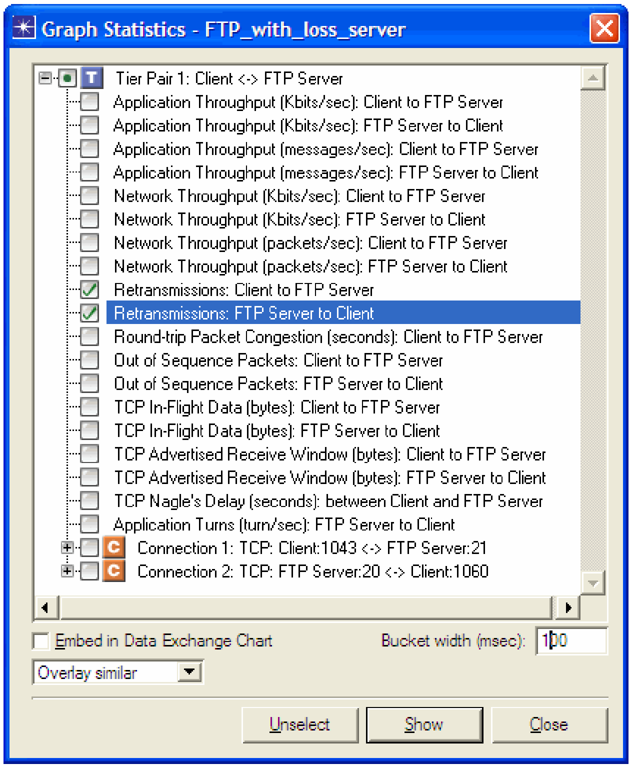
The graph shows the network throughput in each direction for the file transfer.

5 Examine the red line, which represents throughput from the FTP Server to the Client, to see that this traffic averages about 300 Kbps and has a spike to about 500 Kbps. But the Frame Relay circuit has an available bandwidth of 768 Kbps. You conclude that something is throttling the flow of this file transfer over the WAN circuit.

Keep the graph window open, but move it out of the way before performing the next step.

6 Return to the Graph Statistics dialog box. Unselect the throughput statistics and select only the two Retransmissions statistics.

7 Change the bucket width to 100 msec.



Graph Statistics - FTP_with_loss_server

- ⊟ ● Tier Pair 1: Client <-> FTP Server
  - ☐ Application Throughput (Kbits/sec): Client to FTP Server
  - ☐ Application Throughput (Kbits/sec): FTP Server to Client
  - ☐ Application Throughput (messages/sec): Client to FTP Server
  - ☐ Application Throughput (messages/sec): FTP Server to Client
  - ☐ Network Throughput (Kbits/sec): Client to FTP Server
  - ☐ Network Throughput (Kbits/sec): FTP Server to Client
  - ☐ Network Throughput (packets/sec): Client to FTP Server
  - ☐ Network Throughput (packets/sec): FTP Server to Client
  - ☑ Retransmissions: Client to FTP Server
  - ☑ Retransmissions: FTP Server to Client
  - ☐ Round-trip Packet Congestion (seconds): Client to FTP Server
  - ☐ Out of Sequence Packets: Client to FTP Server
  - ☐ Out of Sequence Packets: FTP Server to Client
  - ☐ TCP In-Flight Data (bytes): Client to FTP Server
  - ☐ TCP In-Flight Data (bytes): FTP Server to Client
  - ☐ TCP Advertised Receive Window (bytes): Client to FTP Server
  - ☐ TCP Advertised Receive Window (bytes): FTP Server to Client
  - ☐ TCP Nagle's Delay (seconds): between Client and FTP Server
  - ☐ Application Turns (turn/sec): FTP Server to Client
  - ⊞ ☐ C Connection 1: TCP: Client:1043 <-> FTP Server:21
  - ⊞ ☐ C Connection 2: TCP: FTP Server:20 <-> Client:1060

☐ Embed in Data Exchange Chart

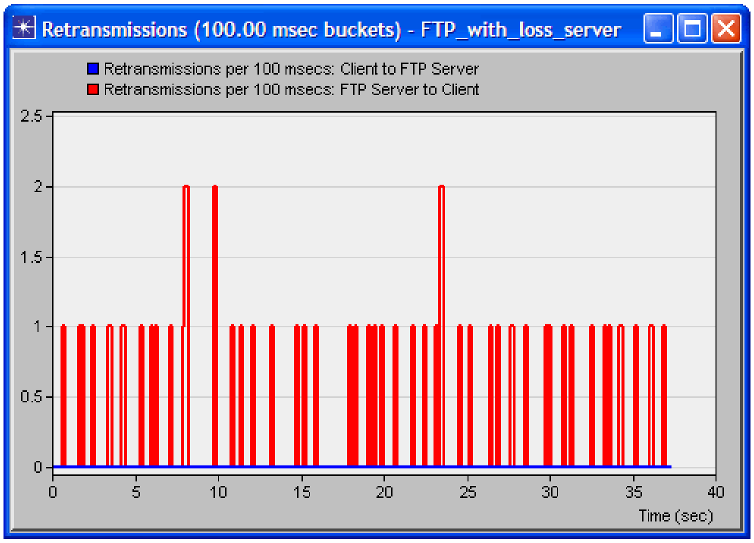
Bucket width (msec): 100

Overlay similar

8 Click **Show**.

➥ The statistics graph for retransmissions appears.

Figure 2-13 Statistic Graph for Retransmissions



9 Examining the graph, you can see when the packet loss that AppDoctor reported occurs. The red line shows retransmissions from the FTP Server to the Client; the blue line shows retransmissions from the Client to the FTP Server. All of the retransmissions occur on the stream of traffic from the FTP Server to the Client. The loss seems to be fairly evenly distributed throughout the application transaction.

Keep the graph window open, but move it out of the way before you do the next step.

10 With this final piece of information, you can summarize what you have learned about the application from AppTransaction Xpert:

- Application flow is experiencing significant throttling.
- Approximately 68 percent of the file download time is due to protocol effects.
- The application is experiencing 4 percent packet loss.
- Packet loss causes TCP to throttle the flow of data across the network.

11 You conclude that packet loss is mainly responsible for the slow file transfer.

12 Close the Graph Statistics dialog box (but keep the two graph windows open).

End of Procedure 2-4

Now continue with the last task, documenting your analysis.

Publish Reports

There are many ways to publish AppTransaction Xpert information, including:

- Capture screens
- Export AppDoctor information to spreadsheets
- Create MS Word (.rtf) reports
- Create web reports

Web reports provide quick access to the information in your current working environment.

Procedure 2-5 Generating a Web Report

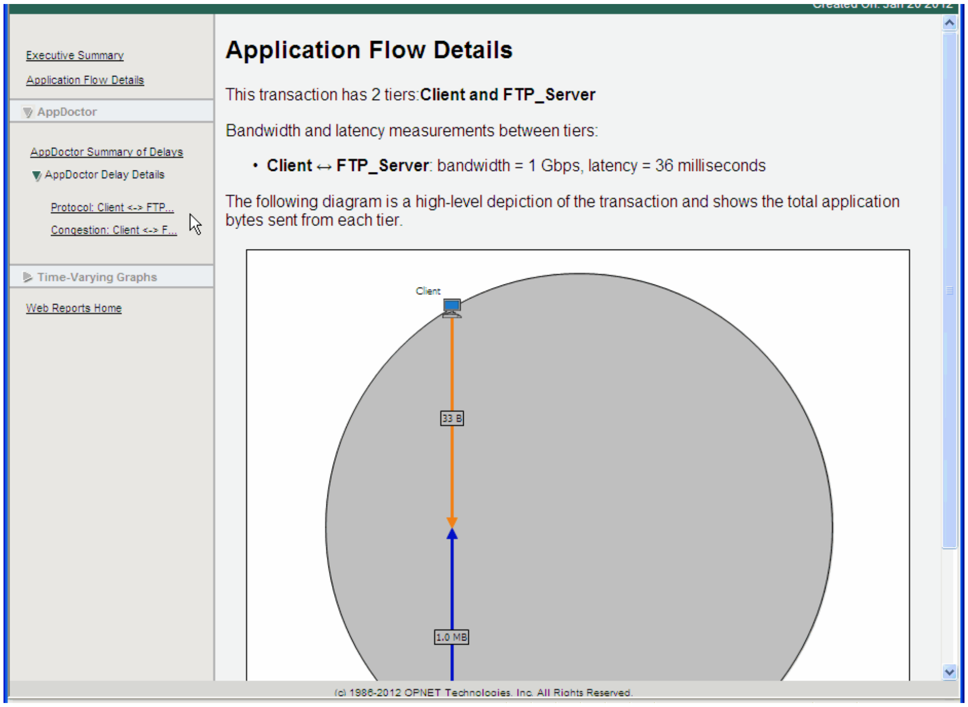
1 Choose **Reports** > **Generate Web Report**…

2 Click **Generate…**, then navigate to the directory to which you want to save the report and click **OK**.

Note—You can specify the location for storing AppTransaction Xpert reports by setting the **ace_web_report_dir** preference.

➥The web report for your current session opens in a web browser.

3 Click on the links in the navigation pane at the left of the browser to view the charts and graphs that were saved from your session.



Application Flow Details

This transaction has 2 tiers:**Client and FTP_Server**

Bandwidth and latency measurements between tiers:

- **Client ↔ FTP_Server**: bandwidth = 1 Gbps, latency = 36 milliseconds

The following diagram is a high-level depiction of the transaction and shows the total application bytes sent from each tier.

(c) 1986-2012 OPNET Technologies, Inc. All Rights Reserved.

End of Procedure 2-5

Conclusion

This tutorial showed how to use AppTransaction Xpert to

- Diagnose a performance problem with a common network application
- Publish and distribute your analysis