14 Data Exchange Chart

The Data Exchange Chart shows the data transferred between tiers on a time line from the start to the end of the application task. You can view the transaction using the following views: Network Packet Chart and Application Message Chart.

The Data Exchange Chart includes the following features and options:

- Application message dependencies—The Application Chart can display dependencies, which describe the network/application delays between message transfers across tiers. The “View Dependencies” check box in the upper-right corner allows you to hide/display dependencies.
- Protocol decodes—The bottom pane shows detailed decodes of packets/messages that are selected in the chart. To show/hide the pane, choose View > Embed Protocol Decode Viewer.
- Statistic Graphs—You can show graphs of statistics that measure network and application performance within each tier pair and connection. You can also import and show statistics generated by external programs, such as AppInternals Xpert, perfmon, and GlancePlus.
- Zoom level—You can adjust a chart’s zoom level at any time by right-clicking in the chart background. The Zoom pop-up menu allows you to zoom in or out; zoom to a defined region (Zoom to Rectangle), selected group of packets (Zoom to Selection), or a tier (Show Single Tier); return the previous zoom level; and zoom out completely.
- Packet/message visualization—The View > Visualizations submenu enables you to visualize characteristics of individual packets/messages such as payload/retransmissions, TCP header types, and HTTP payload data.
- Packet/message tooltips—To view tooltips for any packet, message, dependency, or packet/message bar, select the item with the mouse and the place the mouse pointer over the selection.

For more information, see:

- Network Packet Chart
- Packet/Message Bars
- Application Message Chart
- Application Message Dependencies

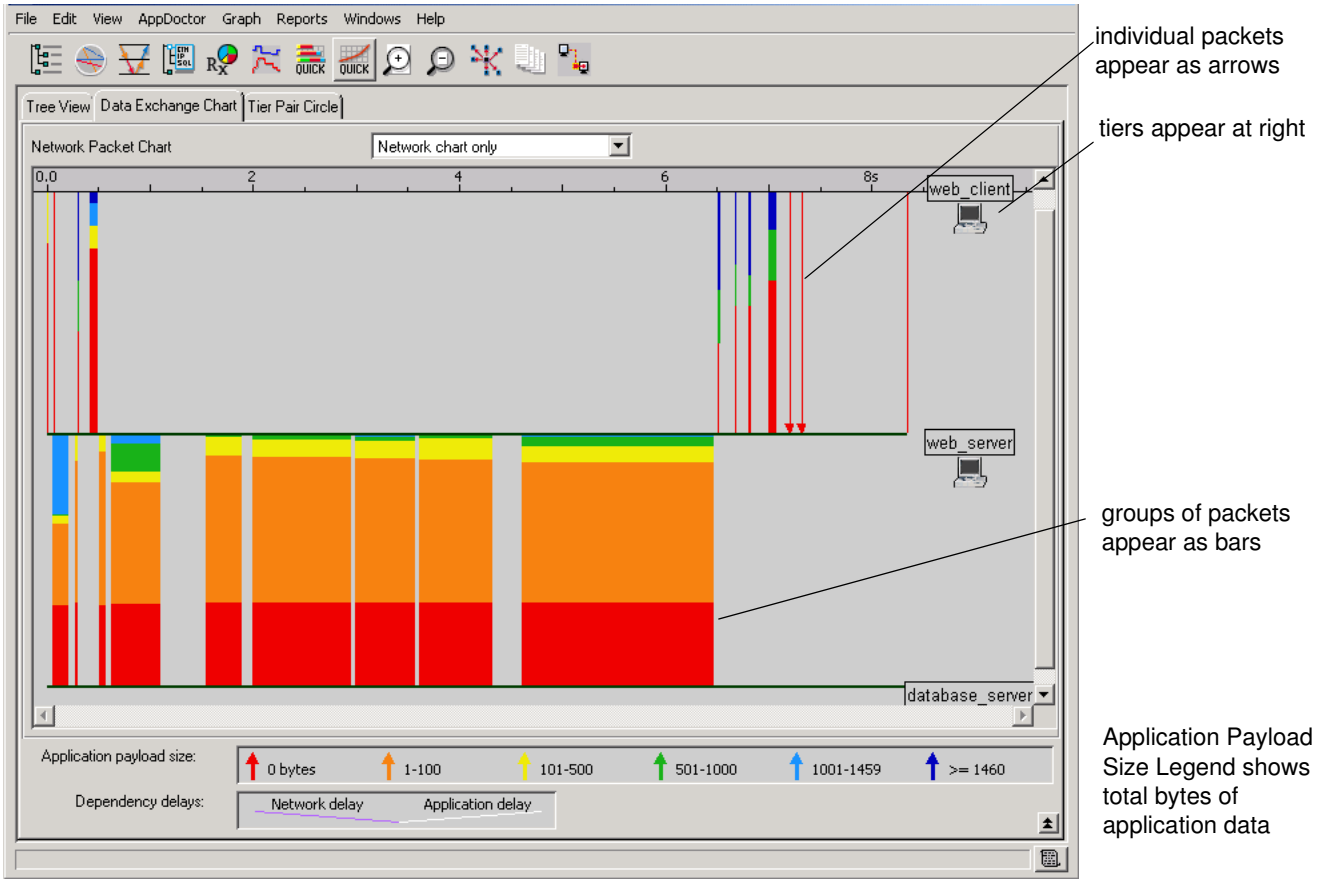
Related Topics

- *Tree View*
- *Tier Pair Circle*

Network Packet Chart

The Network Packet Chart, shown in the following figure, arranges packets sequentially from the start of the recorded exchange to the end, from left to right.

Figure 14-1 Network Packet Chart



Each packet arrow and packet-group bar is color-coded to indicate the amount of *application-layer* data stored in the corresponding packet(s), as shown in the legend below the chart. The amount of color in each bar indicates the proportion of packets that fall into each application-payload size category, as explained in section Packet/Message Bars.

The following table lists the elements of the Data Exchange Chart, shown above.

Table 14-1 Elements of Data Exchange Chart

Item	Description
View pull-down menu	Selects the Data Exchange Chart view: • Application chart only • Network chart only • Application and Network charts

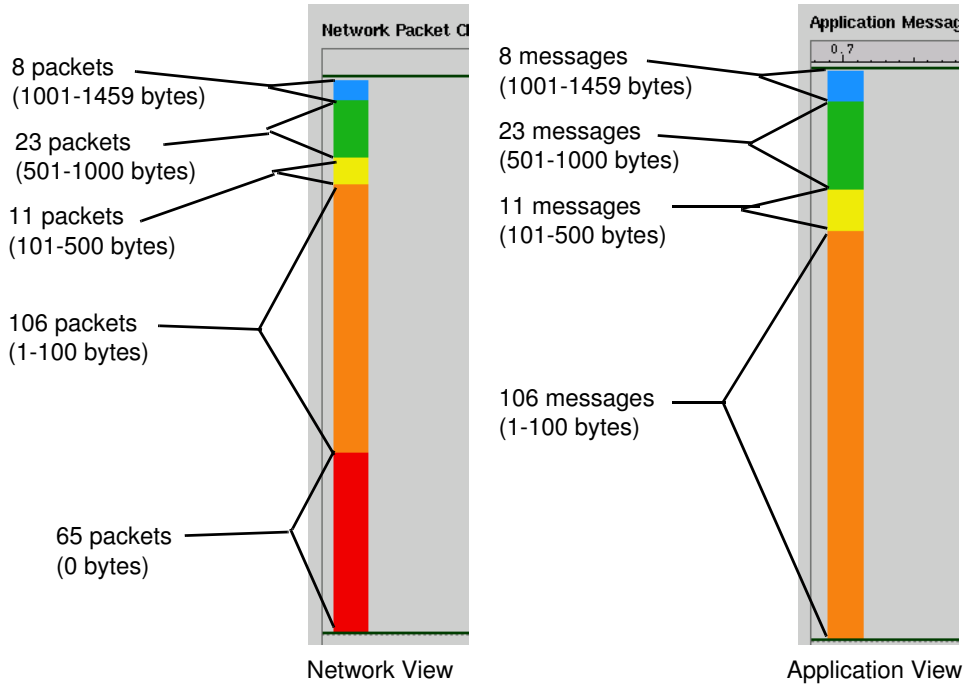
Table 14-1 Elements of Data Exchange Chart (Continued)

Item	Description
Time ruler	This ruler indicates the x axis of time in seconds from the start (left) to the end (right). Note the following options: • To change the time scale between relative and absolute times, choose View > Time Scale. • To zoom to a specific time range, choose View > Set Visible Time Range.
Network packet arrows	The chart represents each packet as an arrow travelling from one tier to another. The horizontal placements of the arrow's head and tail indicate the packet's arrival time (at the destination tier) and departure time (from the source tier), respectively. Each arrow is colored by the amount of application-layer (no total) data contained in that packet, as indicated by the Application Payload Size legend below the chart.
Tier direction arrowheads	The Data Exchange Chart uses gray triangles or trapezoids at the top and/or bottom of each group to indicate that the group contains packets being transmitted to the top and/or bottom tier.
Network packet groups	If packets are grouped too closely in time to be shown as individual arrows, the chart represents the packet clusters as solid bars. (See Data Exchange Chart for a description of these groups.)
Tier names	The tier name appears to the right of each tier line.
Tier lines	Each tier in the application is represented by a horizontal line.

Packet/Message Bars

Packet or message bars are color-coded to indicate the packet or message size. The figure below interprets the color coding.

Figure 14-2 Packet/Message Bars in the Data Exchange Chart

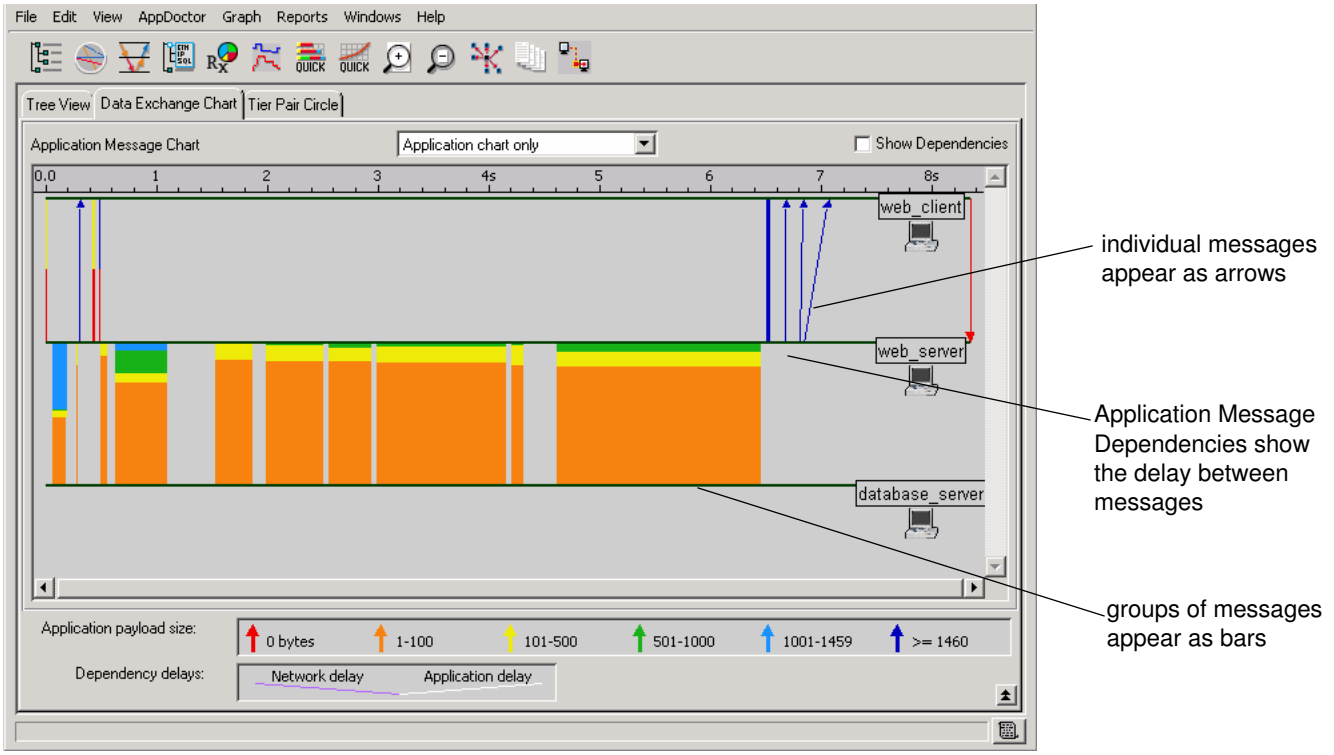


Application Message Chart

The Application Message Chart uses the same conventions as the Network Packet Chart, with the following exceptions:

- Arrows and bars represent application messages, not network packets.
- The Application Message Chart can show dependencies, which describe the cause-and-effect relationship between different message transfers and their associated delays.

Figure 14-3 Application Message Chart



The following table lists the elements unique to the Application Message Chart, shown above.

Table 14-2 Elements of Application Message Chart

Item	Description
Dependency lines	Dependencies are divided into two components: • Network delay (shown as a purple line) • Application delay (shown as a white line) For more information, see Application Message Dependencies.
Show Dependencies checkbox	Specifies whether to show/hide dependencies in the Application Message Chart.
Application message arrows	This chart uses colored arrows to represent application messages instead of network packets, and uses the same coloring scheme to indicate the amount of application-layer data in each message
Application message bars	If messages are clustered too closely in time to be represented by individual arrows, the chart uses a solid bar to represent the cluster of messages.

Application Message Dependencies

Dependencies describe the causal relationships and delay times between application messages. Viewing dependencies in the Application Message Chart can help you pinpoint situations in a data exchange where network and/or delays are impairing your application's performance.

A Transaction Analyzer model typically describes an application-level transaction composed of sequential phases. We might describe our example task as follows:

1) The client sends a data request to the web server;

2) The web server forwards the request to the database server;

3) The database server retrieves the requested data and returns it to the web server;

4) The web server formats the requested data as an HTML page and sends it to the client.

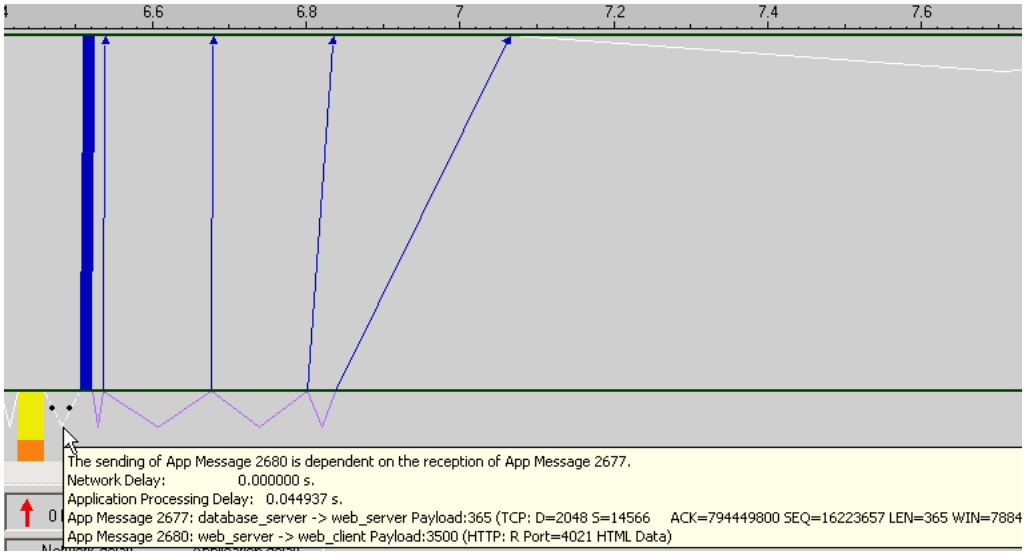
Clearly, each action depends on completion of the previous event. The web server must receive a complete request before it can query the database; hence the web server's sending of the first database query message depends on (i.e., is a dependency of) its receipt of the last client request message.

Dependencies have two important characteristics:

- Each dependency occurs at a particular tier;
- Each dependency has an associated wait time, which represents the delay between the "cause" and "effect" events. Each wait time is split into separate components of network and application delay times. The network delay represents transmission, propagation and protocol delays incurred when transmitting the message. The application delay represents the interval when an application is processing—*not* transmitting or receiving—data.

You can view message dependencies in the Application Message Chart by checking the View Dependencies checkbox in the upper-right corner. Each dependency is composed of two lines that represent the proportion of network delay (purple line) and application delay (white line). You can view a tooltip summary for a dependency by placing the mouse pointer over one of these lines, as shown in the following figure.

Figure 14-4 Message Dependencies in the Application Message Chart



Dependencies make it easy to identify the individual elements of delay in a task and provide important information about the application's performance. Two facts are apparent from the chart above:

- Nearly all of the dependencies (and delays) occur during exchanges between the web server and the database.
- Nearly all of the dependencies are composed primarily of application, not network, delays. This suggests that, to reduce the response time for this type of request, we should examine how the web server communicates with the database server.

Dependencies and protocol decodes, used together, provide a powerful tool for analyzing application performance. You can use dependencies to pinpoint large application processing delays and then use the Protocol Decodes operation to examine in detail the application messages that trigger the delays.