How "Distributed by Design" Maximizes Observer® Bandwidth Efficiency

Summary

This document explains how Observer products are designed to efficiently use network bandwidth when deployed in a distributed environment.

Keywords

Advanced Probe, Distributed, bandwidth, network trending, compression, encryption

Introduction

Network Instruments' Console/Probe "Distributed by Design" architecture lets you deploy economical Probes where ever you need a point of visibility from which to analyze traffic. Network Instruments' Probes aren't just easy on your financial budget; they're easy on your "bandwidth budget" as well.

Consoles and Probes

The Observer display console is exactly that: it *displays* the data that is captured and *pre-processed* before being sent by a Probe. Depending on what you are doing with Observer, there are three flavors of pre-processing:

- When in statistical modes such as Bandwidth Utilization or Top Talkers, the Probe only sends the data points necessary to update the statistical listing or chart. The amount of data transmitted varies depending on the statistical display. For example, Top Talkers requires station addresses as well as packet, byte, and utilization statistics, while Bandwidth Utilization requires only utilization statistics.
- While capturing packets or network trending data, the Probe sends updates on the status of the capture or trending updates, until you click the View icon to display the decode window or the Network Trending Viewer.
- When you click the View icon to view remote packet captures or network trending data, the Probe compresses and encrypts the data before sending it to the console. In the case of trending, it's not the actual packet data being compressed, encrypted, and transmitted, but rather the database of sampled network metrics that you have chosen to track.

In the first two cases, the Console acts as a "thin client" to the Probe (where the bulk of processing occurs). When viewing actual packets or trending data, the Probe compresses and transmits the data to the Console for analysis and display.

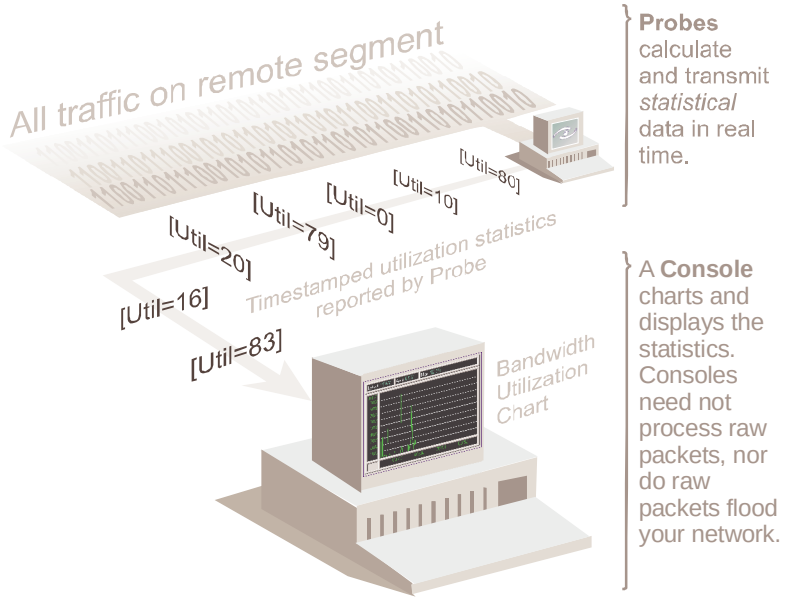
Network Instruments, LLC
8800 West Highway Seven
Fourth Floor
Minneapolis, MN 55426 US
(952) 932-9899
(952) 932-9545 Fax

www.networkinstruments.com

Observer Statistical Displays

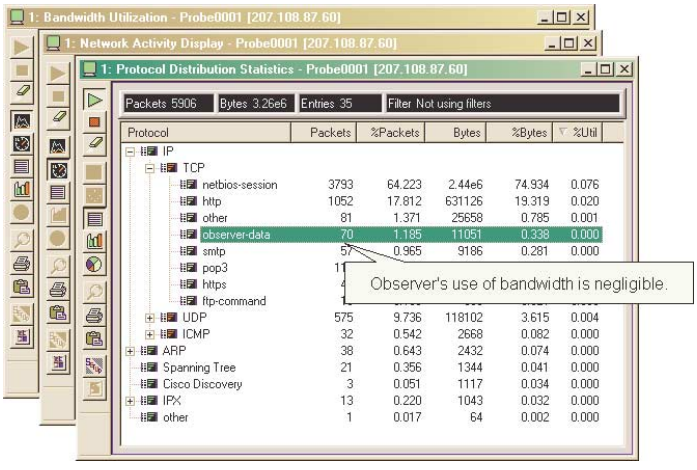
Observer's distributed architecture doesn't just give you the necessary "points of visibility" in a multi-segmented network—it also limits bandwidth and processor overhead. The data that drives Observer's statistical displays (such as Bandwidth Utilization and Top Talkers) is the result of analysis performed on the Probe—the Observer console doesn't have to process remote packets, and your network doesn't have to deliver them.

How Observer Statistical Displays are "Distributed by Design"

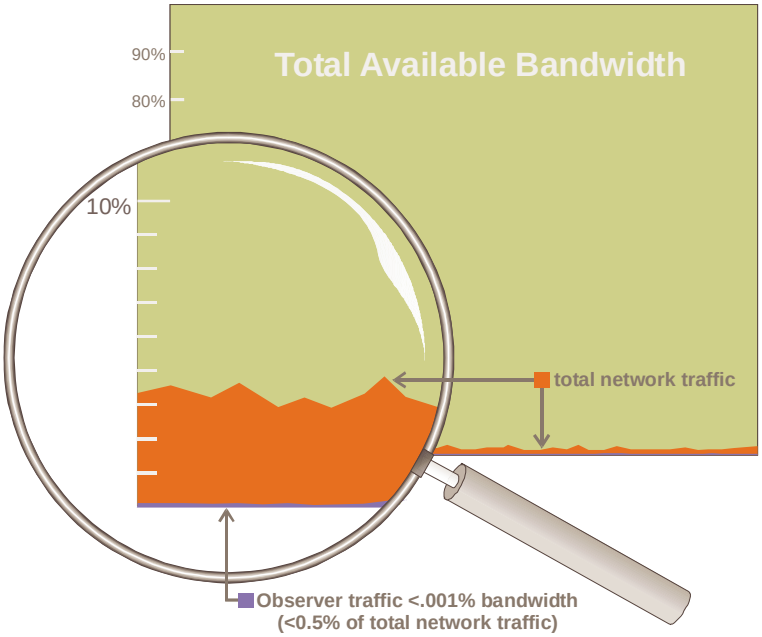


For example, monitoring a remote segment's bandwidth utilization will cost you a small packet or two every couple of seconds. Top Talkers might cost you a few more, depending on the number of stations and traffic patterns on the segment.

You can of course use Observer to introspectively analyze itself. The following screen was taken from a live network. It shows Protocol Distribution statistics after receiving Bandwidth Utilization, Network Activity, and Protocol Distribution Statistics from a remote Probe for a few refresh cycles:



As you can see when charted, the Observer traffic barely registers:

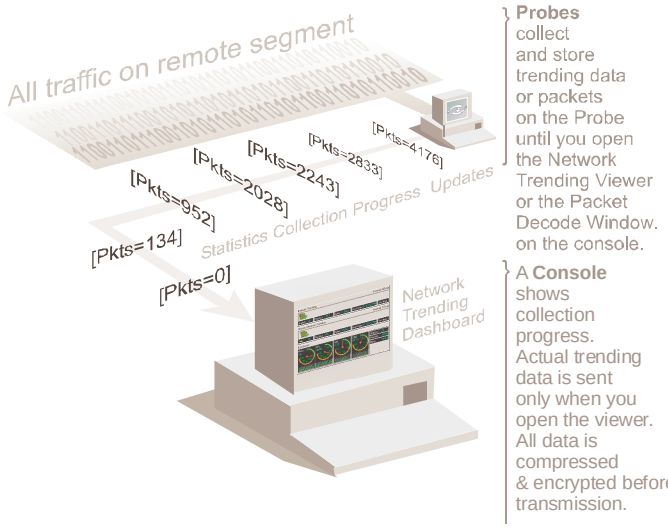


This is typical. Note that even if your bandwidth utilization is higher than this, Observer traffic *does not increase proportionally.* If for some reason you need to reduce Observer's bandwidth usage even more, you can always set the Probe's report period interval to a higher value(**Options->Select Probe or Device Properties**), or reduce the number of statistical displays you run simultaneously.

Packet Capture/Network Trending Displays

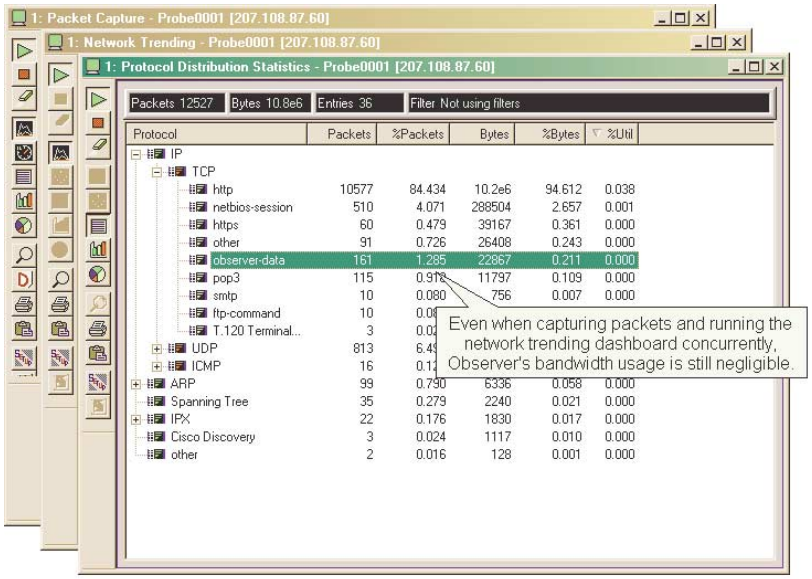
Starting remote capture or trending (e.g., by clicking **Capture->Packet Capture** or **Trending/Analysis->Network Trending**), Observer displays only the current *status* of the remote data capture until you click the View icon.

Status Updates for Trending and Packet Capture



The Packet Capture graph and the Network Trending dashboard will typically generate a packet or two every refresh cycle to keep their

displays updated. Let's look at another Protocol Distribution display from a live network, in this example running with both Packet Capture Graph View and Network Trending Dashboard activated:

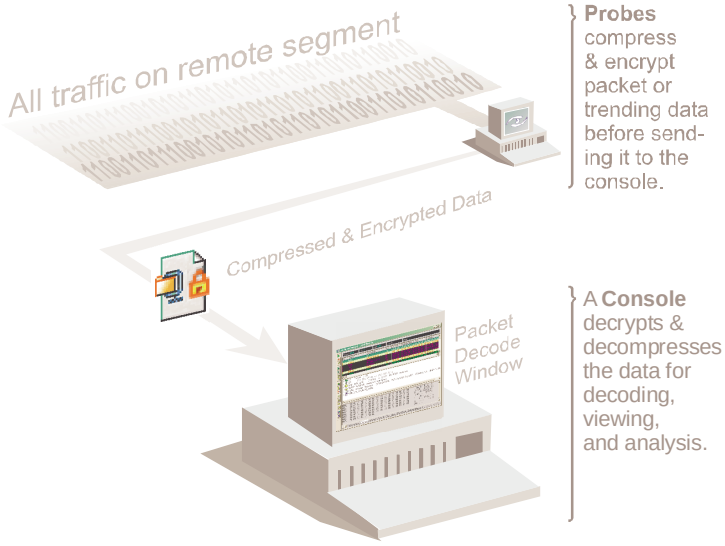


As with the other statistical displays, the Network Trending dashboard and packet capture graphs are updated on the same refresh interval set in the Probe or Device Properties dialog.

Packet Decode/Network Trending Viewer Displays

There's no getting around it—when you want to look at the actual packets or network trending data, all of it has to be shipped to the console. Whether it's a lot of raw packets or a large chunk of Network Trending statistics, Observer minimizes the impact on your network by having the Probe compress the data before it is transmitted.

Decode and Network Trending Viewer Data Transfers

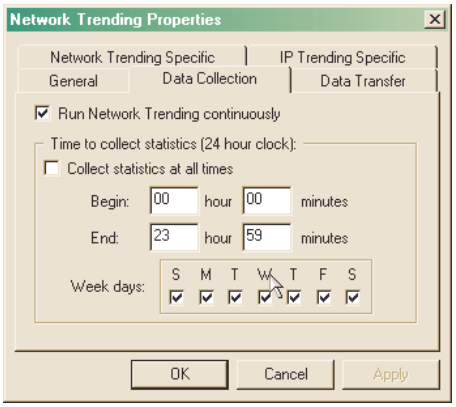


The proprietary compression algorithm not only reduces the size of the data transfer by up to 85%, it also acts as a low-level encryption

scheme, masking the content of the packet capture data during transmission.

When capturing packets, you can decrease the amount of data to transfer by applying filters during the capture rather than post-filtering the packets after transfer.

With Network Trending, you can schedule fewer and shorter data collection periods if you want to reduce data transfers. Click the Setup button on the Network Trending Dashboard. The Data Collection tab lets you schedule Network Trending to the periods of time you are interested in:



Conclusion

You buy an analyzer to *solve* network problems, not to add problems by flooding your network with analyzer traffic. Observer's "Distributed by Design" architecture minimizes the analyzer's impact on your network by processing data remotely where possible, and using compression whenever it must send "raw" data. This not only saves bandwidth, it allows an Observer console to monitor statistics from multiple remote Probes simultaneously, as most of the processing is distributed to the Probes.