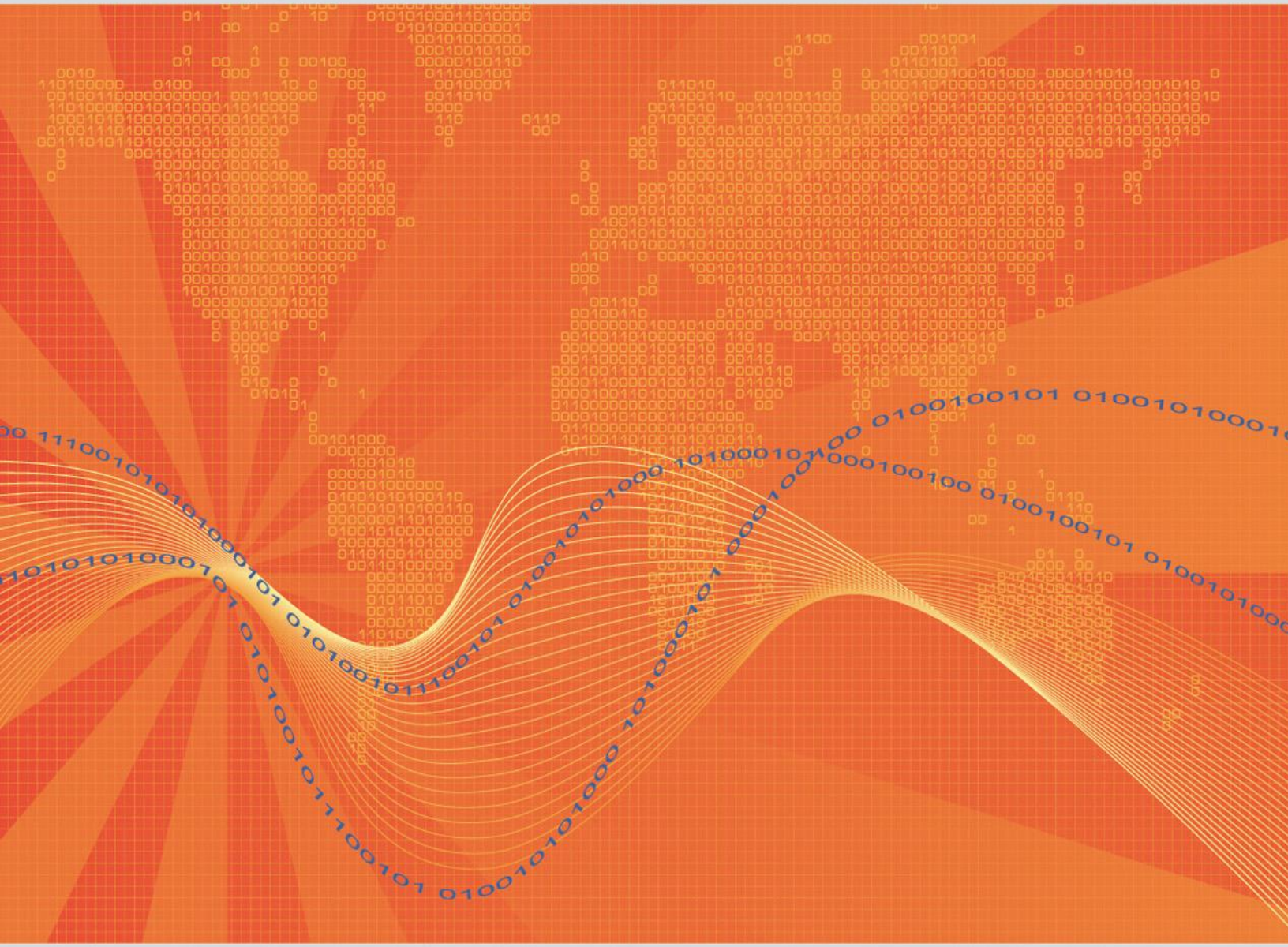
Application Note:
The SecureLinx Spider Network

A Guide to Maximizing Distributed KVM Installations



Lantronix, Inc.
15353 Barranca Parkway
Irvine, CA 92618
Tel: +1 (800) 422-7055
Fax: +1 (949)450-7232
www.lantronix.com

Contents

Introduction

As the Lantronix SecureLinx™ Spider, a KVM (keyboard, video, mouse)-over-IP solution, gains popularity in the distributed IT environment, several questions have arisen with regard to its implementation, limitations, and performance. As an ideal solution for the distributed IT/ remote branch office environments, Spider offers networking concepts and protocols that allow for seamless integration into any network.

This application note provides the information needed (protocol settings, bandwidth and performance, scalability, etc.) to ensure your Spider is working at an optimal level in your network.

Key Concepts

The Spider Difference

Unlike traditional KVM switches, Spider offers a flexible, scalable, and affordable CAT5-based remote access KVM solution in a cable-friendly, compact "zero-footprint" package.

Unlike traditional KVM solutions, Spider provides continuous availability to servers with 1:1 non-blocked, BIOS-level access. This allows administrators to have guaranteed access to mission-critical servers regardless of how many of them need remote access. In other words, administrators are not "locked in" to a fixed number of remote users. Spider offers an extremely low-cost-per-remote-user for guaranteed non-blocked access. No client software or an external power supply is required.

The Spider boots in approximately one minute upon plugging into a server or auxiliary power supply, and no additional hardware initialization or selection time is necessary upon initiating a user session.

The small device is light enough to be suspended by its connection cables behind the target or conveniently stowed within the server's rack. Up to eight client sessions with any single target are possible, though only one client will have the ability to control the target at any time.

Protocols

Typical A/V (audio/visual) streaming protocols such as Real Time Streaming Protocol (RTSP) use User Datagram Protocol (UDP) allow packet broadcast and multicasting with minimal messaging in applications that can withstand dropped, duplicated, or erroneous packets. Spider uses the Remote Framebuffer (RFB) over TCP instead, ensuring that the transmitted packets are reliable and ordered through ACK messages sent from the client back to the Spider. RFB is designed to update all, or a portion of a screen snapshot efficiently using relatively simple compression rather than a timer that triggers the transmission of data over the network.

Working at the framebuffer level, the RFB protocol allows interoperability among the vast majority of clients and servers. In the absence of application-specific software, the client and server simply agree on the protocol version and then can begin transmitting

data. Since RFB does not use motion vector compression, it is not optimized for motion video in the way that MPEG does.

Bandwidth and Performance
Based on user mouse movements and keyboard entries, the client sends cursor updates and ACK messages to the Spider. Emulating a locally attached mouse and keyboard, the Spider provides this information to the target server, updating the screen.

Client requests and ACK messages require minimal compression with a 60 byte, whereas large target screen updates with over 1,500 bytes per packet will require more data compression over the TCP layer and greater processing time at the client. Thus data flow through the Spider is expected to be asymmetrical, with greater bandwidth requirements going upstream, from the Spider to client, then downstream from the client to the target.

While the Spider's dedicated processor is continuously capturing, digitizing, and compressing the video from the target system, it does not stream data onto the network until a client initiates a session or requests an update. A Spider without an active client session uses no network bandwidth and is invisible to the network. Thus it is the number of simultaneous sessions rather than the number of Spiders installed that limits the network bandwidth.

Scalability
Each Spider has an Ethernet port and a Cascade port. With the Cascade port, Spiders may be linked together to minimize the number of cables. The Ethernet chain is switched rather than shared, so there is fundamentally no difference between the Spider being linked to the Cascade port or connected directly to a switch through the Ethernet jack. The chain operates at 100 Mbps, so latency is inconsequential. With single point of failure, maintenance, and accessibility, the number of Spiders in a chain is inconsequential; however, up to 16 or 32 devices can be supported without noticeable congestion.

Simulating Interfaces

Each Spider installation will be unique to the environment, topology, and traffic of its network, but it is useful to have baseline figures in order to maximize KVM installations within a bandwidth budget. To provide quantitative performance data, simulations were performed for typical and heavy interface traffic with a single target and single client connected through one Spider.

Setup
A Windows XP target was connected to a Spider. The Spider was accessed with a client also running Windows XP over an otherwise unloaded 100 Mbps subnet. Wineshark, a popular open source network analysis tool, was used to capture all packets on the subnet.

	Target	**Client**
Operating system	Windows XP	Windows XP
Processor	550 MHz P3	2.6 GHz P4
Resolution	1280 x 1024	-----
Video Card	-----	Nvidia GeForce FX

The Spider settings on firmware v2.1 are:

Interfaces → KVM Console Settings → Transmission Encoding set to LAN (high color)

Interfaces → Keyboard/Mouse → Host Interface set to USB

Interfaces → Video → Filter set to Normal

After opening the remote console window and pressing the "Video Auto-Adjustment" button a couple of times, the Spider continued to show 0 Bps "In" and 0 Bps "Out" when inactive. Wineshark verified the accuracy of the reported activity level.

Typical Loading

Typical user activity was simulated by opening and closing windows, traversing directory trees, and opening and closing applications repeatedly over a five minute period. The rate of activity was high but not frenetic. Finely detailed Windows XP "Paradise" wallpaper was displayed on the target such that closing an application or window would result in redrawing the wallpaper at every pixel in that area. Since the simulation focused on performing actions that would result in major screen changes, this test is considered the upper bound of data flow with a user performing real work on the system.

This simulation was performed twice and produced consistent results. Graphs of activity for each second of the simulation can be seen in Appendix A. The data below represent the average of the two repetitions.

5 Minutes Simulated Activity		
Total count, each direction	**Packets**	**Bytes**
From Spider	29K	23M
To Spider	22K	1.3M
Averages, bidirectional total		
Packets per second	200 pps	
Mbits/second	0.65 Mbps	

Heavy Loading

In addition to the application simulation, a second test was run with motion video to characterize the capacity of the Spider's compression engine and protocol stack. (Note, Spider's refresh rate is not designed to watch full screen video.) A one minute movie clip with a large amount of detail and action was run on the target system at full screen size in order to force continual updates of every pixel. Though the target system was not particularly fast, it was capable of displaying this full screen motion video without obvious jerkiness. This clip was replayed for a period of five minutes and transmitted to the client system via the Spider. Detailed information can be found in Appendix B and summarized in the table below.

5 Minutes Motion Video		
Total count, each direction	**Packets**	**Bytes**
From Spider	242K	270M
To Spider	120K	6.5M
Averages, bidirectional total		
Packets per second	1200 pps	
Mbits/second	7.3 Mbps	

Summary
Typical user interaction with a target server results in an average network load of 650 kbps under these simulation conditions. Even though the packet count in each direction is similar, the average upstream data rate over the full five minute test is more than 10 times the downstream rate. An average network load of 7.3 Mbps was achieved in the case of video transmission from the target to the client.

Appendices A and B illustrate bursts of simulated user activity data. The scale factor of the graph resulted in truncation of several peaks, though the maximum observed activity in any one second interval was approximately 12 megabits. The movie data rates are more evenly distributed around the mean. One segment of low update activity and the lag time appear consistently for all five iterations.

Achieving Maximal Performance

On a local network, the round-trip transmission time was small enough that the user input and screen updates were synchronized. Even a large number of concurrent Spider sessions would be unlikely to cause significant congestion on a 100 Mbps local area network. However, this network setting is not ideal in the typical distributed environment. The following guidelines can be used to assess the capacity of a Spider-based network, reduce the impact of each Spider session, and maximize KVM accessibility for multiple simultaneous users.

Network Capacity
Based on the results of the simulations and the network specifications, it is possible to calculate how many simultaneous Spider sessions can be supported. Capacity is not limited by the number of Spiders at the site, but rather the number of active, simultaneous users, their level of activity, and how sensitive each is to performance.

Each user logged on to a Spider is counted as a session, including multiple users connected to a single Spider. Even though the same target server screen is displayed on the client, the data rate must be multiplied by the number of sessions to quantify the actual network impact of all active Spider sessions. It may be useful to fill in a table to track and budget the available bandwidth. An example format is shown below.

Subnet: _____ Bandwidth Up: _____ Bandwidth Down: _____				
Consumer	Simultaneous Quantity	Bandwidth Required (Up/Down Mbps)	Total Bandwidth (Up/Down Mbps)	Remaining Bandwidth (Up/Down Mbps)
Regular traffic	----			
Static Spider Session				
Typical Spider Session				
Heavy Spider Session				

If you desire support for more simultaneous sessions, you can adjust the Spider's settings based on performance requirements of each user.

Latency and Bandwidth
Mouse-to-cursor performance is sensitive to latency, network congestion, and TCP retransmits. On a congested or slow network, the round-trip time for screen updates can

become large enough to cause mouse and cursor tracking problems or, when using the PS/2 mouse connection option, asynchronous data transfer.

According to the simulation, network bandwidth averages 650 kbps per session when there is a vigorous amount of user activity. This means that clients using DSL or cable modem connections should not experience compression issues. However, Spiders that connect to DSL or a cable modem will be limited in the number of possible external connections. The upstream rate needs to be considered along with the faster downstream connection rate advertised for the DSL or cable modem service. This could be a particular disadvantage if multiple Spiders, accessed by multiple users, are connected to DSL or a cable modem. While the bandwidth dictates the cost of service for most Wide Area Network (WAN) connections, the availability of upstream bandwidth can alleviate simultaneous session constraints.

If the target screen is changing rapidly enough to saturate the upstream connection, cursor lag at the client may result. Wireless network users are particularly susceptible to multiple TCP retries due to congestion downstream from the Spider. Cellular data networks, including UMTS (Universal Mobile Telecommunications Systems) and EV-DO (Evolution-Data Optimized), can also have significant latency issues.

If your Spider is configured similar to the simulation, the network may experience degraded performance. Hence, if smooth cursor feedback is required, the above sessions may benefit from high-compression settings.

To change those default settings in firmware v2.1, navigate to Interfaces → KVM Console Settings → Transmission Encoding

You can also make changes to the compression settings during a session by navigating to Options → Encoding → Compression. More compression means less data is sent over the network.

Static Images
In many instances, a client initiates a Spider session to run scripts or copy files, which results in minimal screen updates or a static image. A Spider with a static image should take up zero or very little network bandwidth; therefore a large number of sessions are possible. For example, a blinking cursor with no other activity should use no more than several hundred bytes per second. If the lower right corner of the remote console window indicates significant data coming from the Spider, then click the "Auto-Adjust" button once or twice to eliminate it. **If the activity levels do not subside, you can navigate the menu (firmware v2.1) to Interfaces → Video → Noise Filter, and select "Large".**

Display Settings
It is possible to change the display settings in order to reduce the amount of data sent in each update, but it is not necessarily obvious which settings will affect the desired change. For example, changing the encoded color depth of the client's display or scaling the remote console window, will not affect the amount of data sent from the Spider and will degrade the image on the client. A client workstation that has six Spiders logged-in, each RC window scaled to 25% is the same as six individual users logged into six Spiders with full-size RC windows. However, in the former scenario the user will only be capable of interacting with and therefore receiving screen updates from one Spider at a time.

The easiest way to reduce the bandwidth requirements is to minimize the target server's screen resolution. While this simulation used a resolution of 1280x1024 pixels to show

maximum bandwidth utilization, reducing it to 800x600 pixels should be fine for most applications. This eliminates 60% of the data sent on each update. Since the RFB protocol is based on events rather than on a temporal element, the refresh rate of the captured display is irrelevant to the network data rate. Selecting the slower 60 Hz setting will consume less power in both the server and the Spider.

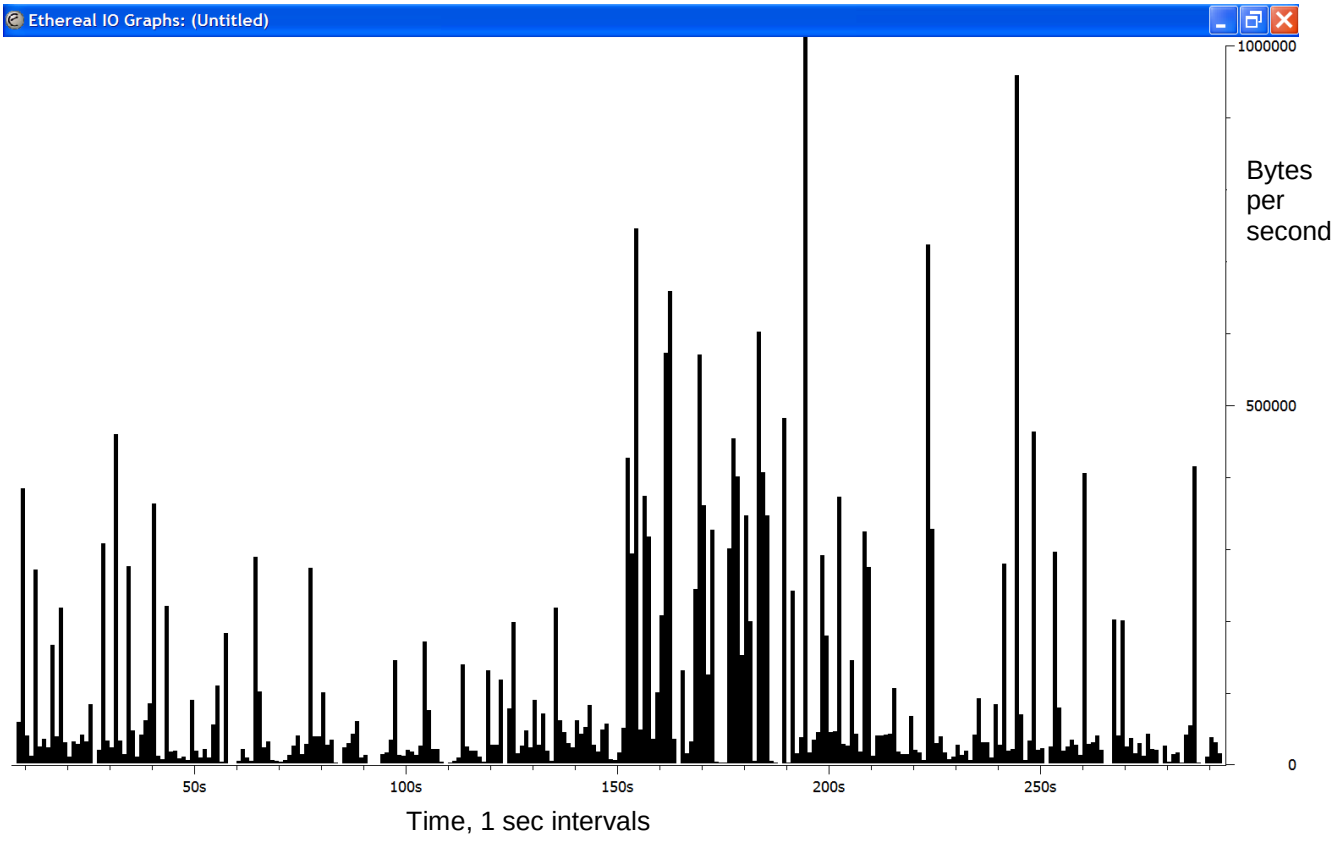
A second method is to change the target's encoded color depth from 16 to 8 bits, cutting half the amount of data sent in every update. **With firmware v2.1, navigate to Interfaces → KVM Console Settings → Transmission Encoding, and change the selected option from "LAN (high color)" to "LAN".** While the reduced color depth may not be aesthetically appealing, no other compression artifacts will be introduced.

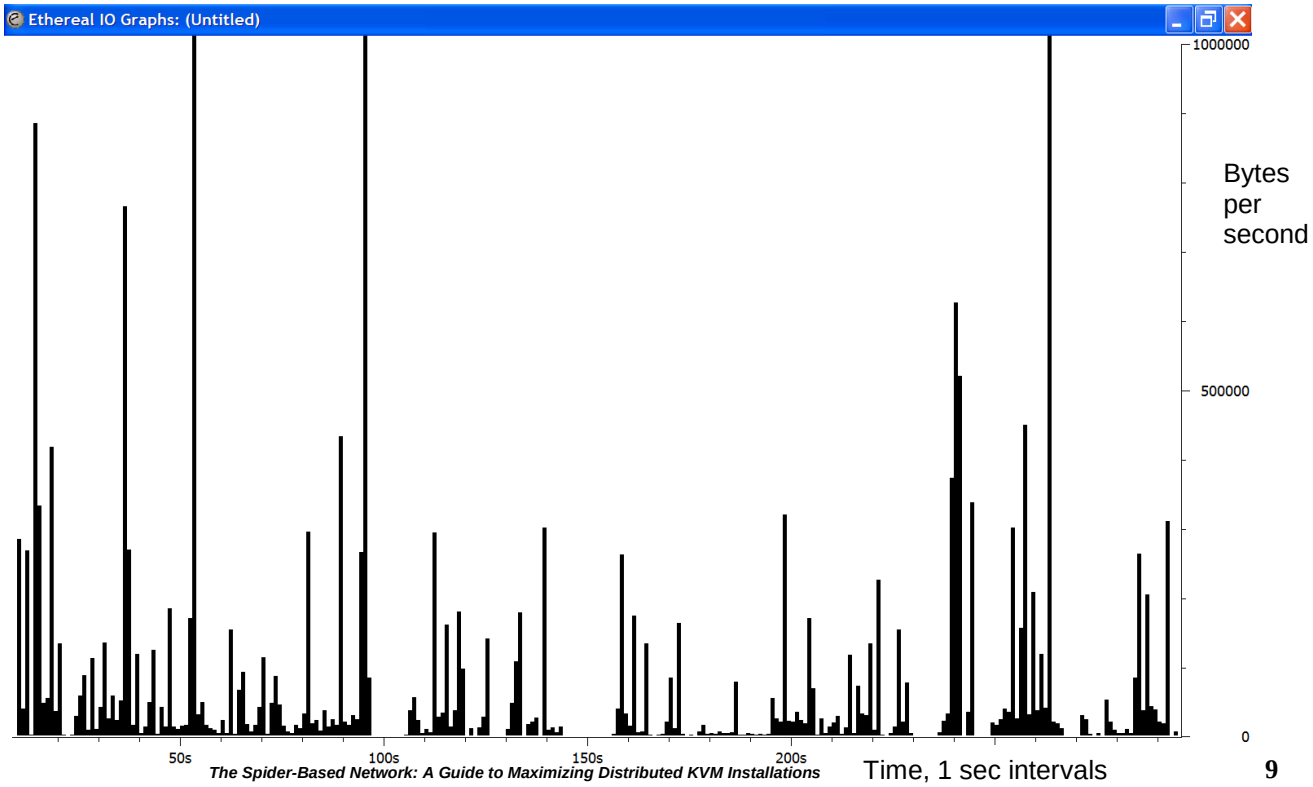
By taking advantage of both methods, the network can support five simultaneous users with the same bandwidth as the original settings. In some settings, an application may require special tuning. Pre-defined encodings for a single user are available for various types of connections. These encoding settings are specific to each login name, so users with different connections can be set to the appropriate default. A little experimentation will reveal the optimal combination for a particular user count and network configuration, and it can be updated based on actual conditions.

Conclusion

Clients connected to the Spider behave like any other client/server system, and the interface is adjustable to suit the needs of any network. The data and suggestions above provide practical information for installation. If special situations arise or problems are encountered, standard network analysis tools can be used to analyze, reconfigure, and troubleshoot. A simulation as simple as capturing and analyzing traffic with Wineshark will provide useful quantitative data for a specific network. Understanding how the key concepts and simulation results apply to real-world network performance criteria will assist greatly in maximizing the simultaneous accessibility of multiple servers in a distributed IT environment.

Appendix A. Spider Data Test Captures – Simulated User Activity (2 Runs)





Appendix B. Spider Data Test Captures – Movie Clip