NOMACHINE		Introduction to NX technology	
Prepared by: Gian Filippo Pinzari		*N°:* D-309/1-NXT-DOC	
Approved by: Gian Filippo Pinzari	*Signature:*	*Date:* 26/09/2003	*Amended:* A

Introduction to NX technology

Index

Introduction to NX technology 3

- Open Standards Drive the Success of the Internet.....3
- Why the X-Window System 3
- X-Window System's Challenger 4
- NoMachine's X Protocol Compression Technology.....4
- OpenSource, a Mission Statement.....5
- Linux as a Server and Desktop Platform 5
- RDP and RFB Foreign Protocols.....5
- NX Distributed Computing Architecture 6
- Distribution of Load Between Nodes 6
- Designed for Simplicity 7

Introduction to NX technology

A brief introduction to NX motivation and technology

This document outlines the background and the design decisions that guided NX development. It explains why NX is different from similar technologies and states the goals the NX project is set to pursue.

Open Standards Drive the Success of the Internet

Network Computing is an old "buzzword" of the Internet. It is the idea of freeing users from reliance on their Personal Computer and Operating System and moving towards computing made of applications and servers. So far, the lack of a universally accepted protocol (like HTML and XML for the Web), has made it difficult to implement and support network computing on a large scale.

Why the X-Window System

NX Distributed Computing Architecture is a suite of Open Source technologies and commercial tools, designed to make network computing as easy and widespread as the Web browsing. It consists of a thin layer of server software that enables any Unix computer to work as a terminal server and clients for a wide range of platforms and operating systems. NoMachine has chosen to build the foundations of its NX Distributed Computing Architecture on the well known and widely used X-Window System, the windowing system that's behind the Graphical User Interfaces of Linux and the Unix Operating System.

Most network computing solutions seem to be designed to work as surrogate tools. It appears that the architects behind their design did not intend them to be the primary means by which users access their desktops. This is the problem, for example, with VNC and RDP. Both these protocols are much simpler than X (and so very well suited for thin clients), but their simplicity does not compensate for their lack of efficiency and functions. These protocols, for example, draw the remote screen by transferring huge amounts of image data over the network. Even if RDP is of a much higher level and a much more efficient protocol than RFB, it has not been designed for everyday use of computing resources, but as an add-on to the underlying operating system.

X-Window is the graphical subsystem ,and not an extension, of the host OS's subsystem. X applications communicate with X-Window using the X protocol, so that the operating system does not have to add a layer to translate screen updates into a network

protocol.

X-Window System's Challenger

X-Window System is an open and extensible client-server protocol, designed to offer a layer of separation between the application logic (running on application servers) and the presentation (taking place at clients). In other words it has been designed for network computing. The native X protocol requires high bandwidth and low latency networks to work at its best. Over the years, though, evolution of Graphical User Interfaces has increased the demand for network and computing resources. Instead of layering X applications to adapt to available network resources, X client and X toolkit designers have relied on users giving up X-Window's networking capabilities to simply access applications running on their own computers. Over the last few years, the X-Window system has completely lost its original characteristics of network computing protocol, being perceived as a mere display driver even by the most informed and technological users.

The main goal of the NX project, since the beginning, has been to develop X compression technology that would enable any user to run unmodified versions of the most widespread X desktop environments on a standard X server, over any type of network connection. This is, according to us, what is first needed to allow X-Window system regain its function of networking protocol.

NoMachine's X Protocol Compression Technology

NoMachine has developed exclusive X protocol compression techniques and an integrated set of proxy agents that make it possible to run complete remote desktop sessions, even at full screen, using narrowband Internet connections, at speeds as low as those offered by a 9600 band modem.

NX compression operates at three levels on the X protocol:

- It compresses the network traffic by a variety of means, including -per message- differential algorithms, advanced caching methods, lossless and lossy image compression.
- It reduces network round-trips nearly to zero, maximizing the throughput.
- Adapts bandwidth in real-time, according to the network conditions.

NX provides X protocol compression ratios ranging from 10:1 to 100:1 and over, depending on the application being displayed. This is obtained without penalizing

performances on fast LANs, so that remote sessions can usually run at a speed that is indistinguishable by a local computer.

Open Source, a Mission Statement

NoMachine has released as Open Source all the core libraries and low level software components developed to support its NX Distributed Computing Architecture. This includes the X agents, the client proxy and all the libraries implementing compressed transport of X protocol. NoMachine has chosen the GNU General Public License, the same license that drives the development of the Linux operating system. NoMachine aims to support a wide community of developers, working together to create interactive server and client software to make the Network Computing revolution possible.

Linux as a Server and Desktop Platform

NX Server products run on Linux Operating System. Linux is a Unix operating system and thus offers all the characteristics of reliability, scalability and performance required by today's corporate environments. At the same time, its flexibility make Linux a good candidate to power the distributed, peer-to-peer network of millions of connected computers providing services and applications, which will constitute the network computing of tomorrow.

Whilst being an established platform in the server operating system market, Linux is also emerging as a reality on the desktop. Desktop environments like KDE and GNOME, and the StarOffice award winning office productivity suite, use the X-Window protocol. NX support of such environments comes at zero cost and provides the best performances overall.

RDP and RFB Foreign Protocols

NX accessibility and remote computing capabilities are not limited to Linux desktops and servers. It is also available for numerous different operating systems. NX encapsulates and translates into X protocol the Remote Desktop Protocol used by Microsoft Windows NT/2000 Terminal Server Edition and Citrix Metaframe. It also translates Remote Frame Buffer, the protocol used by VNC, another Open Source remote computing facility.

Although NX compression offers the best performances when running native X applications, RDP and RFB sessions can be compressed by a factor ranging from 2 to 10. NX support of foreign protocols provides further advantages. Firstly, it extends its reach to virtually any computer and secondly, NX offers to the user a unified view of any application

resource available over the Internet.

NX Distributed Computing Architecture

NX Distributed Computing Architecture is designed from scratch for distribution of workload between nodes in a Wide Area Network. The NX Servers are intended to perform high-level, manageability functions on the NX network. They are responsible, for example, for authentication and activation of user sessions, as well as for ensuring that nodes export their familiar computing environment to users (for example the same file systems and the same applications). Sessions run on NX Node computers in a virtual cluster. Hundreds of NX nodes can be connected to one or more controlling servers to support thousands of concurrent sessions.

NX Nodes are not intended to be dedicated computers (even if this is likely to be a common scenario in legacy contexts). Any computer on any desktop can be a node of a NX Distributed Network. An NX network is very similar to a peer-to-peer network where clients search for a song (in this case a single application or a desktop environment) and the search engine redirects the client to the appropriate server, providing, on behalf of the client, the authorization credential that will have to be submitted to let the node accept the connection.

Distribution of Load Between Nodes

Not only can servers balance the network load by distributing sessions among nodes, but nodes can distribute the applications running in a session between different application servers. This can happen either because a resource or an application is not available on the original node, or because a resource is "closer" to the user and so the shorter network path guarantees better performances.

A session can be identified with an agent running on a node. Agents are available for X, RDP and RFB network protocols. Agents run in proximity of application servers and accept connections from NX clients. Clients are the users' computers where X Server runs. Only one (socket) connection to the X server is needed by each agent. The case of RDP and RFB is straightforward. These protocols emulate a complete screen and do not require communication between different applications connected to the same X server. A node can connect any RDP and RFB sessions to a different Windows Terminal Server computer or a different VNC server, balancing the load of sessions among the available network resources.

In the case of X protocol, two operating modes are possible: when a user chooses

to run a session in a new virtual desktop, the agent itself acts as an X server. It multiplexes X client connections into a single connection to the real X server. X applications can be run on different nodes, acting as application servers. Communication between X applications is resolved locally, on the side of the agent. As this communication can be very verbose, the overall performances are greatly improved. At the moment X applications run in X agent appear to be "nested" inside the agent's window. A "rootless" agent, that is an agent overcoming this limitation, is currently under development.

When a user chooses to run applications inside the existing X session, windows are "floating" on the desktop and integrate perfectly with local applications. In this case, NX operates as a transparent proxy, compressing X traffic and tunnelling the add-on network services (like multimedia and SMB protocol) but without offering local resolution of X protocol's round-trips, made possible by NX agent.

Designed for Simplicity

In spite of the apparent complexity, the built-in networking capability of Unix and the inner characteristics of scalability of X-Window and Unix network services make it straightforward. NX uses SSH remote execution facilities to gain access to a node's functions. All the communication takes place encrypted through industry standard SSL public-key cryptography. No new network servers need to be installed on nodes, except the SSH daemon, a server that comes standard in any modern Unix distribution.