

Thin Client Computing With The Tektronix/NCD XP/NC 200

Mark A. Davis 4/2006

Introduction/History

“Thin Client Computing” refers to shifting the running of software to a server and using a lightweight client on the desktop to display and interact with that software application. Thin clients have been around forever. Some of the first thin clients were designed to run the X Window System under Unix, to bring the sophistication of graphical applications to the desktop.

In the early days of thin client computing, cost of CPU power, memory, and hard drive storage made thin computing a necessity. The first thin clients were teletypes- a printer with a keyboard attached. Later, CRT terminals came on the scene, using a monitor to display text-only, attached through serial ports on mainframes. Various types of terminals followed, some supporting limited graphics, others with much faster network connections. Xterminals were introduced to bring a full and rich environment to users.

As time went on, “fat clients” started appearing in the early/mid 1980's with the introduction of microcomputers. It finally became common to have fat clients in the late 80's/early 90's. However, businesses often were confronted with a myriad of unexpected problems with fat clients- high initial cost, and even higher “total cost ownership”. Fat clients were of little use to businesses unless they were connected together with networks. Suddenly they were faced with not only having to maintain servers and data centers, but also networks and scores of individual machines. Each machine having a separate storage system to maintain, install, and backup. Security was often a nightmare- viruses, trojans, removable storage, stolen machines, etc.

There was a backlash against fat clients in the mid 1990's, and many businesses started installing modern thin clients to regain control of information and skyrocketing costs. Of course, thin clients are not right for every application, so mixed networks of thin, medium, and fat clients were common. Citrix invented thin client computing for MS-Windows with the ICA protocol. Even Microsoft got into thin clients by copying the way X worked by creating RDP (remote display protocol). Both had only small success because they put extremely high prices on the licensing enabling such connections- the user has to pay for total simultaneous connections, plus each POSSIBLE (not active) display on the server.

Xterminals enjoyed success throughout the 1990's. It wasn't until the 2000's that their popularity started failing due to ultra-cheap X86 computers and free OS's such as Linux and BSD. It was finally possible to build thin clients using off-the-shelf components and paying no licensing to anyone. Or, you can create a medium client, one that runs some local clients and also some remote clients. Also, using Linux/BSD, it is easy to turn old hardware into thin clients for free, although they will never quite be a small, sleek or power miser as true Xterminals.

Advantages of using Xterminals

- 1) The user is logging into a remote server, so the user can use their login on any machine on the network and have all their applications, files, and permissions right there, instantly.
- 2) It is much easier for the IT department to manage Xterminals than fat clients.
- 3) There is no data stored on Xterminals, so there is nothing to lose, backup, or have stolen. Security is much easier with terminals.
- 4) There are no expensive applications or OS on the terminals to purchase, install, or maintain.
- 5) Xterminals use much less electricity than fat clients.
- 6) Xterminals create much less heat.
- 7) Thin clients often have no fans, and rarely spinning storage, so they create zero noise.
- 8) Thin clients are much more reliable than fat clients- no moving parts to fail and less overall parts.
- 9) Xterminals are physically small, taking up less in-use space and less storage space. They are easier to install, remove, or move.
- 10) Thin clients typically boot much faster than fat clients.

Disadvantages of Xterminals

- 1) They place a much higher load on the servers and network than fat clients.
- 2) They only operate well with a low-latency network.
- 3) They don't support multimedia (animation/sound) well, because of the nature of thin client computing.
- 4) They are often difficult or impossible to upgrade the hardware or have connected peripherals.
- 5) The user usually perceives slower application speed than local clients.
- 6) If the server or network fails, the users can't do anything at all!

Tektronix Xterminals

Tektronix made a variety of Xterminals/thin clients. The ones used in this presentation are the XP200 or NC200 series. The NC is typically much faster than the XP, even though they look identical from the outside. They can “speak” the X or ICA protocols. They have a very small ROM-based OS and a type of “BIOS” that allows the user to configure aspects of the terminal such as its address and where to boot from. They have a small amount of flash memory to permanently store settings. The power supply is external and the network connection is a standard RJ jack for 10baseT ethernet (should work OK on a 100baseT network with appropriate hub/switch).

Limitations of XP/NC 200's

- 1) Tektronix sold its thin client division to rival NCD, and later, NCD went out of business. So there is no official support for the devices anymore.
- 2) The XP/NC is 8 bit pseudo-color (256 colors), not all modern applications like running in such modes. They have to compete for and share colors. If one application is “not nice” and steals all the colors, other applications will look strange, since they will run with incorrect colors.
- 3) The XP/NC has poor memory management, so its memory can become fragmented over time. Although the applications run on the host, not the terminal, the terminal still needs memory for display purposes and running its tiny OS. A memory shortage manifests itself as applications not launching or terminating abnormally.
- 4) Modern gtk-linked applications are, for some reason, incompatible with the old Xserver that runs on the XP/NC. This includes recent versions of firefox, openoffice, and other gnome-desktop-included applications.
- 5) They do not support Xcinerama, so trying to use them as additional, contiguous displays off your main console is not possible.
- 6) With some KDE versions, kdm will crash the xterminal before the login screen is shown. Use xdm or no session manager.

Uses of the XP/NC 200's

So what can I DO with this thing?” Assuming you can use non-recent GTK linked applications, they are great for:

- 1) Fun/experimentation/learning.
- 2) Coding/programming.
- 3) System management- especially if your console graphics card gets messed up
- 4) System monitoring- you can have CPU load and other indicators on a nice screen
- 5) Web browsing spot- with the exception of sound, you can browse the web pretty effectively
- 6) Multiuser your home system- allow someone else to access your machine. Great for checking Email in another room.
- 7) Play some non-intense multiuser games like Xblast, Risk, or Monopoly.

Servers

There are several “servers” on the network that will support the running of Xterminals. They can each be different machines, all be the same machine, or some mixture:

- 1) Address server: to provide IP addresses (DHCP or bootp on old systems)
- 2) Boot server: to provide the OS (through NFS and/or TFTP)
- 3) Font server: to provide fonts to the Xterminal
- 4) Session server: to provide the login screen and authentication (XDMCP)
- 5) Application hosts: to run the actual X applications

Address Server

Typically, Xterminals use DHCP to get their address, name, and default route. On old systems, you can use BOOTP, but that is rarely used nowadays. This server is not really necessary, since you can hard-code the XP/NC's address in the BIOS if you like. However, part of what DHCP provides is a way to send the boot file to the terminal. If you don't use DHCP, then you will HAVE to use tftp or bootp to send the boot file. Let's assume you WILL use a dhcpd service on your server. You will need to edit the /etc/dhcpd.conf file to add the Xterminal. The boot file is different for the XP200 and the NC200:

```
authoritative;
ddns-update-style interim;
subnet 192.168.1.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
```

```

option domain-name "homenetwork.org";
option domain-name-servers me.homenetwork.org;
option routers 192.168.1.1;
group {
  filename "/tftpboot/tekxp/boot/os.500";
  next-server 192.168.1.1;
  host frog      { hardware ethernet 08:00:11:03:F0:36; fixed-address 192.168.1.155; }
}
}

```

In the example, the default internet gateway is 192.168.1.1, the name of the network is “homenetwork.org”, and DNS server for the network is “me.homenetwork.org”. The terminal is named “frog” and given the address 192.168.1.155. For the XP200, replace the “os.500” with “os.350”. That's about it... just start or restart the service (“service dhcpd restart”)

Boot Server

Once the terminal knows its address, it will need its boot files from the boot server. These are the files that are provided by the NCBridge software CD that normally live in /tftpboot. With DHCP, that file will be sent to the terminal. But if you are not using DHCP, you must setup tftp (not covered in this guide). After the OS file is loaded, it will have to load other files that are not provided by DHCP. The easiest way to get them to the terminal is with NFS. With NFS, it is rather easy- just copy all the NCBridge files to /tftpboot, and add a line like this on your server, in /etc/exports:

```
/tftpboot *(ro)
```

That will allow any machine on the network read-only access to the /tftpboot directory structure. If you are already running NFS, you will then need to “exportfs /tftpboot”. Make sure you “service nfs start” and “service portmap start” if you aren't already running NFS services on your network.

Font Server

Applications require fonts, and they will request them from the font server as needed. “xfs” is the X Font Server. Most Linux distros are either not running xfs, or they are running it for LOCAL use only. You will need to run a font server that is available on the standard port 7100. The configuration for xfs is in /etc/X11/fs/config. But normally xfs is started as a service in /etc/init.d/xfs. Check your script and look for something like this:

```
daemon --check xfs xfs -port -1 -daemon -droppriv -user xfs
```

As you can see, it is set to launch xfs with port “-1”, which is LOCAL use only. The easiest thing to do is to just add another line that will launch xfs for general use, like this, right below the above line:

```
/usr/bin/X11/xfs -port 7100 &
```

Session Server

The terminal runs an Xserver after it boots and uses XDMCP (X Display Manager Control Protocol) to manage the sessions. Essentially, it “asks” the server you configure in the BIOS and/or asks the entire network for someone to provide session service. Typically, you will want to enable XDMCP on your host if you want to see a login screen on the terminal when it is done booting. But that isn't absolutely necessary. You can just use the Xterminal as an extension of an existing session or as a display-only device by setting your DISPLAY environment to the Xterminal's name and launching some Xprogram.

The traditional XDMCP server is “xdm”. But newer display managers such as “kdm” from KDE and “gdm” from Gnome will also work just fine, if configured correctly. For this example, we will enable remote XDMCP with xdm (For kdm, you will need to edit the /etc/kde/kdm file (/usr/share/config/kdm/kdmrc on older systems, look for [Xdmcp] and change “Enable=false” to “Enable=true”. Restart kdm). Edit /etc/X11/xdm/xdm-config and comment out the line “DisplayManager.requestPort: 0” with a “!” in front. That's it. Start xdm and a login screen will likely appear on both your local machine and the Xterminal.

If your default session is KDE, keep in mind that KDE is rather “huge” and color-hungry. Using kdm/kde is going to result in a rather slow session. If you find that this is not the desktop environment you want for the Xterminal, then it might be better to setup icewm as your session. You can control this through the /etc/X11/Session file.

Application Server

Well, this is pretty much automatic. Whatever host provides the session, will automatically launch applications on itself for the user. You can launch applications on other hosts by using ssh or a combination of rsh with exporting the "DISPLAY" environment.

Non-Xterminal Thin Clients

Turning a Linux computer into a pseudo-thin client is extremely easy. Just stop running any existing Xserver (in Mandriva- "service dm stop", procedure varies in other distros). Then launch a new Xserver that requests session management from another machine on your network that is running kdm/gdm/xdm (and configured for remote XDMCP):

```
X -query MACHINENAME
```

That's it! You should then see a login screen from the remote host in a few seconds. If you want that machine to do it all the time, then put the command in your inittab (/etc/inittab), something like this:

```
x2:5:respawn:/usr/bin/X11/X -query MACHINENAME
```

Questions

"Great! It works! But there is no sound..." Well, XP/NC models don't support sound, so that isn't going to work. Even if they did support sound, it gets complicated, since the applications are not running on the terminal, they are running on the host.

"Hey- I tried to run this game and it is REALLY slow." Well what do you expect? First, the XP/NC isn't terribly fast at rendering. Plus, EVERY BIT of screen change has to be pushed over the network... that takes time. If you want fast- then change to running XDM and IceWM desktop.

"What if I want to use this with but without logging in". You can do that by not running xdm/kdm/gdm to them. The terminal will come up with just the root Xweave window, you can then redirect the display of applications from your server to that machine by playing with the DISPLAY environment variable.

"It doesn't work :(I can't boot or I can't get a login screen" Look at the messages on the terminal display, and in /var/log/messages.

"It seems to work but this application won't start" Not all applications can run in 8bit color. Plus, some modern GTK applications are linked with libraries that are incompatible with the version of the Xserver running on the Xterminal. Also, the NC/XP don't support GL at all, so if the application wants to do anything 3D, it will fail.

"Can I expand the memory?" XP: Yes, just get a SIMM with the correct specs. NC: Yes, but only if you can find the funky microdim that is used in those units (good luck).

"16 or 32 bit color?" Eeek. As they are, no. If you can find the proprietary video memory expansion board, then yes.

"Resolution?" The NC200 supports various resolutions and monitor timings. The lowest res is 640x480@60Hz and the highest is 1360x1024@70Hz. Typical mode is 1024x768 @ 76Hz, which will work with any monitor from the past decade, including most LCD panels. This is the default mode (11) that all the terminals were set to. You can use the "monset" command in the BIOS to tweak those settings. For example "BOOT> se monset 24" then "BOOT> nvsave" will change to 1280x1024@72Hz. See the included PDF manual.

"Can I use some other keyboard or mouse?" Yes, they are both standard PS2, so most devices will work, even wireless, but don't expect scroll-mouse support.

"USB?" Nope. "Serial connections?" Yes, it is possible, but I have never tried it.

"I have more than one machine I would like to log into. Possible?" Yes, by using xdm's "chooser". In such a case, you tell the Xterminal's BIOS to allow all connections, it will ask for services over XDMCP and then display a graphical menu of all servers you can connect to. Click on the one you want and a session will start with that server.

"Can I have more than one Xterminal on my network? How many?" There is no artificial limit to how many you can have. You can add as many as you like. The practical limit depends on how many are in use at the same time and which applications are being used. There are many other factors (type of network, switches or hubs, backbone speed, server CPU and RAM, etc).