Seizing control of someone’s computer might seem like a violating act of aggression, but sometimes it’s necessary. In the course of a decade of Linux Format, we’ve known people to run remote desktop sessions against the machine they’re actually using (it sounds crazy, but there was more than one desktop running), so don’t imagine you can’t find some use for it.

While real Unixers may like to sing songs about SSH and the command line, you may need to gain access with a graphical desktop, and that’s where remote desktops come in. For many people, the tried and tested method is VNC, and it features in a lot of clients in this test. But there are other protocols and types of desktop.

In fact, the growing vogue among these clients is to support multiple protocols, so whatever desktop you’re targeting, and whichever server it’s running, there should be something suitable here. Obviously, over the course of these tests, we sometimes weren’t running exactly like for like – but that gives a fair test of the differing protocols too. For example, although NoMachine NX supports VNC connections, we tested its performance against its own NX server, which seems to make sense.

We didn’t test the TightVNC client, mainly because it’s very close to the TigerVNC implementation – they have a common codebase, but TigerVNC has a few more features.

“Whatever desktop or server you’re targeting, there should be something suitable here.”
Remote desktop Roundup

Krdc
The client half of the KDE remote desktop.

Since we gave Vinagre the opportunity to work with Vino, its Gnome compatriot, we thought we’d use a standard KDE desktop on the client and server side and try Krdc with the Krfb server. Somehow, even though it’s implementing the same VNC protocols that everything else does, this combination is about the worst thing after TeamViewer in terms of responsiveness. It worked much better with the standard VNC server and Vino than with Krfb.

A less than auspicious start, but wait: Krdc is actually pretty good. Aside from the NX clients and Remmina, it was the only client on which we stood a chance of surviving a round of Armegatron.

The responsiveness and frame rate were great, even if Krdc did still suffer the same background redraw problems as other VNC clients. If your viewing needs change, it has an easy button to switch between full and scaled viewing modes. There are tabbed views for multiple connections, and the panel on the right, rather like Vinagre, also displays a list of bookmarks, recent connections and servers discovered on the local LAN.

There are plenty of settings for the client itself, but a disappointing set of choices for configuring the connection – you get the choice of high-, medium- or low-speed connections, and the software works out which features to use from there. We found this a bit annoying and limiting.

Aside from that, using Krdc was trouble-free, and it also supports the RDP protocols used for Windows remote access. If someone adds an NX plugin, it could become even more useful. If you run KDE and need an occasional VNC client, there’s no compelling reason to change.

The responsiveness and frame rate were great, even if Krdc did still suffer the same background redraw problems as other VNC clients. If your viewing needs change, it has an easy button to switch between full and scaled viewing modes. There are tabbed views for multiple connections, and the panel on

RealVNC Java Client
Can’t be bothered installing anything? Grab a Java client...

This tiny Java-only client can be downloaded from the RealVNC site, but it’s also contained in the RealVNC server software itself. Navigate to the correct port in your browser and the app will download and run, assuming you have Java set up properly. If you want to build it from source, the Makefile is a bit outdated, so you’ll have to edit it and substitute javac for jikes.

You wouldn’t really expect a Java client to top the performance stakes when it comes to something graphically intensive, and this client did little to change that perception. Despite the fact that we couldn’t manage to coerce it out of 256-colour mode for the duration of testing, it still managed to crawl along. The only reason we didn’t spot more glitches on the display was because we weren’t really sure what murky-dither patterns were intentional.

Needless to say, the other options available are pretty shabby, and we couldn’t get encryption to work at all (which is probably a good thing considering the speeds we managed). It comes across as a faithful replica of the native RealVNC client. The TigerVNC client is a fork of the VNC code, so more or less comprises the bits from RealVNC and TightVNC, although development has continued on these.

So, it resembles a slightly less pleasant version of TigerVNC and doesn’t perform very well. If it had sound support, it would probably swear at you and tell you how rubbish you are. However, in an emergency, a Java client is a good standby. You might not need to be able to see everything perfectly to perform a server-saving operation, so it’s worth knowing about.

It supports the RDP protocols for Windows remote access.”

The quality isn’t great, but the bandwidth it uses is low.

“It resembles a less pleasant TigerVNC and doesn’t perform well.”

www.tuxradar.com
Remote desktop

Remmina

The GTK does-it-all option looks tempting.

Though not be the officially Gnome-endorsed client, Remmina certainly looks at home on the Gnome desktop with its GTK stylings. The well-designed layout works just as well on a small notebook as a giant desktop monitor. A minimalist main display contains a toolbar and a list of available connections. Once connected, a new window spawns showing the remote desktop. Multiple connections are managed by easy-to-navigate, named tabs.

The useful toolbar controls are visible all the time, enabling you to rescale the display quickly to fit the available space, go full-screen or even individually control the horizontal and vertical scale of the window. Unlike some of the clients, you don’t get very fine-grained control over the protocol options such as compression, but it does at least give you a choice of colour modes and the four-step quality control, which seems to be a reasonable way of managing bandwidth and CPU use.

Managing connections and bookmarking them is intuitive, although there’s no automatic discovery.

In the responsiveness stakes, Remmina managed to wow us with its performance during the Armegatron test – not only was it playable, it was barely distinguishable from running the game locally, except for a slight (but crucial) delay relaying keyboard taps.

There are no chat or file transfer facilities for basic VNC connections, but these are available for RDP links, which are also supported by this client.

As we were finishing off this Roundup, version 0.8 of the software was released. Among the new features is support for NX sessions! This makes Remmina the client with the widest range of protocol support, to top it off.

“The design works well, whatever size of screen you have to work with.”

TeamViewer

Remote server-based shenanigans, but it offers a lot of features.

TeamViewer is quite a big name in the world of Windows and the software has many major corporate clients, but it’s little used or considered on Linux. Linux support has been in beta for some time, and the software only runs with the help of Wine. It does work though, and offers features beyond the usual Linux clients.

The first advantage, and in some ways disadvantage, is that this client uses a proprietary protocol that enables clients to link up through a central server, which manages a connection from one site to another. An advantage is that, with a variety of clients on offer, you can view a remote system from practically anywhere, even on locked-down systems that wouldn’t allow an SSH connection, or from behind corporate firewalls. Also, it runs on Windows and Mac, so it’s an easy way to cater for all desktops.

The quality of the connection is poor though. It can render a nice desktop at a slow frame rate, or an unreadable desktop at something approaching a nice speed. However, there are added benefits from the proprietary protocol. It can manage audio (badly) and there’s a little chat client, file transfer and some form of VoIP service. We were unable to get the latter working. Connections are managed by dishing out a PIN on one machine, and the user at the other end typing it in, which isn’t as secure as its authors may want you to believe.

There’s a free version for non-commercial use, although you get an annoying, repetitive popup. While it has some feature ideas that would be worth implementing in the next generation of Linux remote desktops, at the moment this is a non-starter for Linux.

“View a remote system from almost anywhere, even behind firewalls.”
Remote desktop Roundup

**TigerVNC**

The one supported by Red Hat.

As soon as you run TigerVNC, you get a good idea of the kind of people who invented it and why. A tiny request pops up and asks for the server you want to connect to – there are no bookmarks, or lists of located servers on the network. If this were a wrench, it wouldn’t be one with a moulded ergonomic grip.

If you click the Options tab though, you’ll find there are plenty of settings – ones relating to the connection and the protocols at least. While the software will automatically select the options for you, you can specify things such as colour depth and compression level if you like. High compression will reduce the bandwidth needed for an effective desktop, at the expense of more CPU cycles being required at both ends.

In use, running against the Vino and standard VNC servers, TigerVNC performed reasonably well. Of the VNC clients, it was much faster than Vinagre, but not quite as responsive as Remmina. There seemed to be a few more refresh problems than most of the other software, with elements of windows shearing off occasionally, and the damaged background not being redrawn for a few seconds. In terms of response though, it was easy to find the cursor (it renders as a dot, even if the display cursor doesn’t keep up) and the keyboard seemed fine. The display is in a single window with scroll bars if it doesn’t fit the local screen – there’s no scaling, other than running full-screen. This client is capable enough, and has obviously been designed mainly for admins, but even so, some rudimentary comforts would have been appreciated. You might be a hardened network engineer, but still appreciate not having to type in IP addresses every time you want to connect to something, after all.

**Verdict**

This shearing effect was a particular problem we had with TigerVNC.

**Rating** 4/10

---

**Vinagre**

Gnome-inspired VNC client that keeps it simple.

Although the name of this app sounds like something you’d put on a salad, you’ll normally find it entitled Remote Desktop Viewer in your Gnome menus, because it’s a standard part of the Gnome desktop. On running, an ordinary-looking window opens. The panel on the right displays discovered servers and any bookmarks. The main part of the window is for the client connection to the server, which can be run full-screen or within the scrollable confines of this window.

If you open multiple sessions with different servers, the remote displays will appear in a series of tabs. The Bookmarks are OK, but can be confusing – there’s little to distinguish them from discovered servers.

Like the other VNC clients on test here, it’s reliant on the server and the features it supports in terms of performance issues. We tried Vinagre with the standard VNC server and with its ‘other half’, Vino. The latter, like Krdp, is a GUI front-end and a VNC server, designed to make it easy to share desktops across computers. We had no trouble connecting to the remote screen, or using the options with the Vino server for features such as JPEG compression or different colour depths.

One curious problem we had was that the cursor often didn’t update on the display very frequently. This doesn’t seem to be an issue with the connection at all, because menus opened and other GUI elements were displayed almost instantaneously. Full-screen mode also proved impossible for us to escape from – the auto-hiding toolbar refused to come out to play, leaving us to yank the rug from under the client to get back to the desktop.

**Verdict**

A nice user interface and plenty of options make this a decent option.

**Rating** 5/10

---

www.tuxradar.com

October 2010 LXF136 | 33
NoMachine NX Client

The non-free, free NX client from NoMachine.

The NoMachine philosophy is quite simple and easy to buy into. Imagine a world where a computer system has such foresight that even its very method of displaying something on the screen is split into a server-client software relationship. Now imagine that after many, many generations of this, the one true way has been lost, and it’s just now a very complicated, overblown display driver. NoMachine wants X to rediscover its greatness, but too much stuff has been added without thought to the server-client relationship. That’s why its approach is completely different to the standard VNC setup.

The NX protocol works over SSH. This brings security and a few other things not native to the RFB protocol used by VNC as standard. It also uses smart methods of encoding and compressing data, and clever use of caches to minimise the bandwidth required. And if you can save the bandwidth, you can do other cunning things with it – why not also ship over the audio feed from the host machine, for example?

An advantage of using the SSH connection is that if you have a user account on the target machine, you can just log in as normal and start a new X session. The corollary is that it’s hard to “grab” a screen that’s already running, although the NX machine can do that through a standard VNC connection instead if necessary, or by launching a shadow session.

Shadow sessions work well, and depending on the setup, the remote machine may have to confirm the connection. For whatever reason, this tends to be markedly slower than spawning your own X session.

Multi-talented tool

As well as VNC, it also supports RPC for Windows machines, with a similar interface, so it can be used as a general remote access tool.

Although the client software makes it easy to configure connections and save them as settings, it doesn’t really give that much love to the desktop users. Somehow you end up with a whole host (intended) of software installed to do a simple job. There are no tabbed views or anything pretty, and you have to set up sessions in a different application than where you launch them from.

But all that goes by the by when you see it run. Whatever magic pixies sit in the pipes pushing the data through, they do an incredible job. NX on NX definitely gives the best user experience, though a lack of server software for Windows and Mac make its application outside the Linux lab a bit limited.

“NX on NX definitely gives the best user experience there is.”

FreeNX

The NX protocol is an open protocol, even though the software produced by NoMachine is proprietary. This means that it’s possible to create a completely open source version of the server as well as the client applications. There have been a few attempts at this, but the most popular solution for those who like to fly the flag for freedom is FreeNX. This is a plugin replacement for OpenNX – it replicates the same executables (bar the client) in the same locations, so it would be tricky to have both installed simultaneously. However, this does mean that any further automation or other clients that use the services can run just as well from FreeNX as they would from the NoMachine version. This Roundup is about testing the clients rather than the servers, but we did get a version of FreeNX up and running, and there was no noticeable difference that we could see.

There are also some standalone clients, OpenNX and QtNX. OpenNX seems to have been abandoned at some point, and although QtNX is used and developed, we couldn’t get it to connect to either of our NX servers – a real shame.

RDP

RDP is the standard for remote desktop access on Windows machines, and the server is built in. The advantage it has over X is that it’s a system designed to run remotely, although it can’t do quite everything that NX can. The NoMachine NX Client and a few others support RDP as a connection method, but it’s rare to find RDP used as a server of choice on a Unix/Linux box. We tried to test against the Xrdp server for Linux, but failed to get it working satisfactorily. The FreeRDP project (http://freerdp.sourceforge.net) looks like the best bet if you need this functionality on Linux.
Remote desktop Roundup

Remote desktop

The verdict

Remmina 10/10

Leaving aside the clients that don’t work that well, there’s almost a war of ideology going on for the top spots. There’s no doubt that NoMachine, although it eschews standard VNC (you can still use it as a VNC client), performs excellently. In the tests, the OpenGL game was fluid and playable. And while it was fiddly to set up, in use it was better than most. The NX protocol may well be the future, but the client software still has a lot to learn from the user experience guys.

The KDE entry, Krdc, performed well, in spite of its disastrous early start with the companion server. That could really be a problem for the Krdc people, because users will likely use them together and be disappointed with the slow performance. If you’re running a KDE desktop and need a simple VNC client though, there’s probably no need to search further. TeamViewer provided an interesting diversion to the main event. It wasn’t anywhere near as responsive as the main contenders, while we had concerns about security and the ‘phone home’ nature of the connection. Having said that, it did layer on extra features. The NX servers do handle sound, but there are some additional options that could be useful when used in a corporate environment.

The winner, by some distance, was Remmina. Performance was exemplary, but that’s not the full story. It had the best feedback and responsiveness of any client, and if you didn’t know better, the remote desktop might have been a normal-speed local machine. More than the performance though, this client had the best mix of features, and a sensible and well-thought-out interface. Although it did split into multiple windows, the controls are always accessible. As well as discovering clients on the network, it has a good way of storing connections, which would still work well if you had to deal with lots of desktops. Support for NX makes this an all-round winner.

“Performance was exemplary, but that’s not the full story.”

Is it necessary to stick with VNC? Should more distros and desktops integrate NX services? Would you rather run RDP? We’d love to hear your opinions and stories – email us at lxf.letters@futurenet.co.uk

Table of features

<table>
<thead>
<tr>
<th>Name</th>
<th>Krdc</th>
<th>NoMachineNX</th>
<th>RealVNC.Java</th>
<th>Remmina</th>
<th>TeamViewer</th>
<th>TigerVNC</th>
<th>Vinagre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4.44</td>
<td>3.4.0-7</td>
<td>4.1</td>
<td>0.8</td>
<td>5.0.8b</td>
<td>1.0.9</td>
<td>2.30.1</td>
</tr>
<tr>
<td>Licence</td>
<td>GPL</td>
<td>Proprietary</td>
<td>Proprietary</td>
<td>GPL</td>
<td>Proprietary</td>
<td>GPL</td>
<td>GPL</td>
</tr>
<tr>
<td>Test memory</td>
<td>16MB</td>
<td>4MB</td>
<td>33MB</td>
<td>14MB</td>
<td>49MB</td>
<td>6MB</td>
<td>12.4MB</td>
</tr>
<tr>
<td>Test CPU usage</td>
<td>0.36</td>
<td>0.06</td>
<td>0.12</td>
<td>0.37</td>
<td>0.08</td>
<td>0.05</td>
<td>0.2</td>
</tr>
<tr>
<td>Test network</td>
<td>2.6MB/s</td>
<td>0.7MB/s</td>
<td>1.1MB/s</td>
<td>2.1MB/s</td>
<td>0.2MB/s</td>
<td>0.52MB/s</td>
<td>0.5MB/s</td>
</tr>
<tr>
<td>Armeegtron wins out of 10</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>VNC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RDP</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>NX</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Image compression</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>n/a</td>
<td>✓</td>
</tr>
<tr>
<td>Tabbed viewing</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Scaling</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Full-screen</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Network browsing</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Bookmarks</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Password management</td>
<td>Kwallet</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>