

GROUP TEST

Tired of hopping from one computer to another, **Mayank Sharma** tests options that allow him to control all his computers remotely.

On Test

Remmina



URL <http://freedp.github.io/Remmina>
VERSION 1.1.1
LICENCE GNU GPL
Can this do-it-all software ward off the competition?

KRDC



URL www.kde.org/applications/internet/krdc
VERSION 4.14.1
LICENCE GNU GPL
Does KDE's default app do enough?

Vinagre



URL <https://wiki.gnome.org/Apps/Vinagre>
VERSION 3.12.2
LICENCE GNU GPL
...does Gnome's?

TightVNC



URL www.tightvnc.com
VERSION 2.7.10
LICENCE GNU GPL
Is the once popular Java app keeping up with the times?

NoMachine NX



URL www.nomachine.com
VERSION 4.3.30
LICENCE Freeware
Is it the best NX client out there?

TeamViewer



URL www.teamviewer.com
VERSION 10.0
License: Freeware
Does it offer more than just convenience?

Remote desktop clients

When you think of remote access the first thing that comes to mind is SSH. System admins have been using it since time immemorial to mount remote directories, back up remote servers, spring-clean remote databases, and even run remote GUI apps. You probably use SSH to interact with your Raspberry Pi anchored behind the TV.

However, there are times when you need to remotely access the complete desktop session rather than just a single app. Perhaps you want to handhold the person on the other end through installing and using a complex piece of graphical software, or want to tweak settings on a Windows machine from the comfort of your Linux distro. That's where remote desktop software comes in handy. Using these nifty little applications you can remotely access and operate a computer from all sorts of devices.

There are various protocols that are designed to interact with a

remote desktop. For this group test we've set up the *Vino* VNC server on a Linux Mint machine and a *TightVNC* server on a Raspberry Pi and on a Windows 8.1 box. Many of the clients on test support multiple protocols. The exceptions are the two proprietary clients, which we'll connect to with their own servers.

A good remote desktop client should be responsive, and we'll rate it higher than a client that does a wonderful job of replicating the remote desktop in true colour but takes ages to register clicks and key presses. We'll also keep an eye out for related features such as the ability to encrypt connections and transfer files and audio along with the remote desktop.

The clients and servers are all running inside our network connected via Wi-Fi. While for maximum performance you'd want them to be connected via Gigabit LAN cables, it rules out the all-important convenience factor for most readers.

“There are times when you need to access the whole desktop, not just one app.”

Protocol soup

VNC or Virtual Network Computing is one of the most popular mechanisms for accessing a remote desktop. At its heart is the RFB (Remote Framebuffer) protocol, which works at the framebuffer level and is therefore supported by all platforms. One big advantage of the protocol is that you can connect to a VNC server with a client from a different

vendor. Then there's Microsoft's proprietary Remote Desktop Protocol (RDP). While the RDP server is only available for the Windows platform, there are clients for Windows, Linux, Mac OS X, Android, iOS and other platforms. Besides these, several proprietary remote desktop solutions have their own proprietary protocols.

Getting started with desktop sharing

The basics behind this essential technique.

A remote desktop sharing session involves a server and a client. The server component is installed on the remote machine that you want to access and the client component is installed on the local machine, or even on a mobile device such as a tablet.

In a typical desktop sharing session, the remote computer (also known as the host, as it's hosting the session) enables a user to view the contents of the host computer's

desktop. The remote machine can host this connection on a local network or even over the internet. Furthermore, the host computer can also hand over control of the keyboard and mouse to the other party. In this case, all keystrokes and mouse clicks on the client are registered on the server as if they were actually performed on the remote machine.

You'll also have to poke holes in the firewall on the remote host machine to make sure it allows the client to connect. Different

remote desktop servers work on different ports. For example, by default the VNC server listens on port 5900 for connections and on port 5800 for download requests.

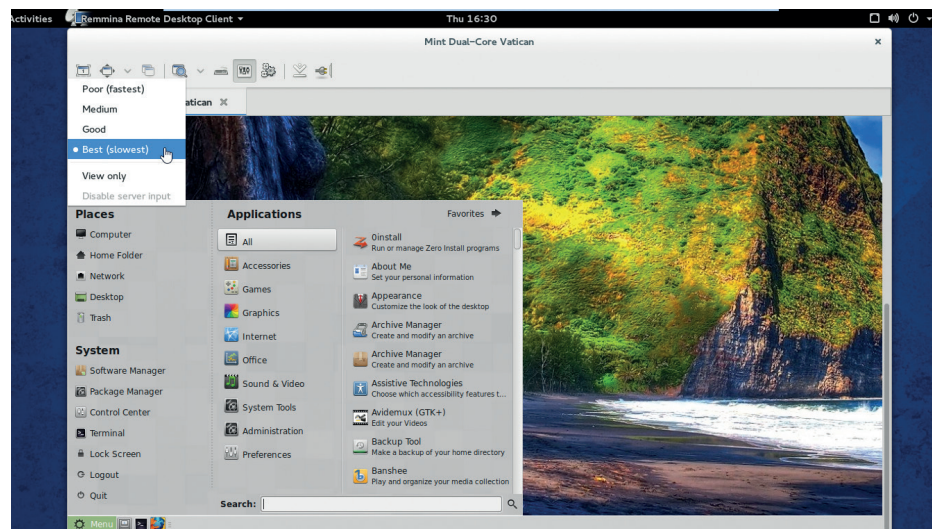
If you use a router, you must configure it to forward connections if you want to connect to a remote desktop over the internet. Remember that in order to establish a remote connection, both the host and the client have to use the same protocol.

Remmina

Maximum performance.

One of the best things about *Remmina* is that the app supports a variety of protocols including VNC, RDP, NX, SSH and more. It has a simple interface, and maintains a list of profiles, and you can organise connections in separate groups. Before you begin, you'll have to create a profile and define the parameters before you can connect to a remote server. At the very least, you'll have to select a protocol from a drop-down list and enter the IP address of the server. Optionally you can define other parameters as well that vary depending on the protocol being used. For example, for VNC connections, you can optionally choose the colour depth and quality of the connection. You also get checkboxes to toggle some quick settings like starting a simple View Only session, disable encryption, and more. If you're connecting to a NX or a RDP server, you can also specify a resolution for the remote desktop.

You can use the app in window mode as well as full-screen mode. *Remmina* has a tabbed interface that enables you manage multiple remote desktop sessions from a single window. When connected you get a bunch of buttons for common tasks such as switching to full-screen mode, or to the scaled mode in case the remote desktop doesn't fit. You also get a button to change the quality of the connection. Unlike other open source apps, *Remmina* changes the colour depth of the remote desktop of the fly, which is a definite plus. There's also a button that sends all the keyboard commands to



Remmina is a *GTK* app and will bring along a lot of baggage when installed on a KDE desktop.

the remote connection. There's a hidden toolbar at the top of the screen which gets you all this control in the full-screen mode.

Remmina houses default remote connection settings under its Preferences window. Here you can tweak some auto save settings for the connections, define default connection resolution and custom hotkeys. RDP users gets a bunch of additional options to help trim down the size of the remote desktop stream, such as the ability to turn off the wallpaper, menu animations, cursor shadow and more.

Stable performer

To test its responsiveness, we tried playing a *Snake*-like game on the remote desktop. When connected to the puny little Raspberry Pi, the game was playable but the keystrokes were delayed by a block or two. Changing to a lower quality level didn't have any noticeable impact on this delay.

Conversely, when connected to a dual-core Mint box with full-HD resolution, the game was playable even at the best quality setting. However, video playback wasn't watchable at any quality setting – at the lowest quality level the video was less jerky but the colours were all wrong; at the other end of the quality setting the colours were perfect but the the video was skipping frames. Also *Remmina* doesn't transfer audio and lacks the ability to transfer files.

We could easily scroll through lightweight PDFs at best quality, while PDFs with lots of images were best scrolled through at lower quality levels and were readable at the lowest setting. *Remmina* is available in the official repositories of most popular distros.

VERDICT

Impresses with its list of supported protocols, features and performance.

★★★★★

Krdc

Krude but effective.

Krdc is KDE's default remote desktop client and supports the VNC and RDP protocols. The app does a nice job of handling connections, with the main interface showing a history of connections with the recently accessed servers at the top. You can also arrange the list by the number of visits to a server. You can even bookmark connections you want to use more often.

Although the main interface might seem overwhelming to a new user, with a handful of menus and buttons, it's fairly simple to operate. To establish a connection, you only need to select a protocol and enter the IP address of the remote machine you wish to connect to.

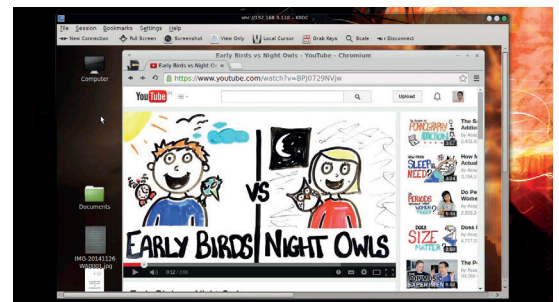
This brings up the host configuration box, from where you have to select a quality setting. The default is Medium, which is claimed to be suitable for DSL, cable, and fast internet connections. There's also high quality for LAN and low quality for slower connections.

And that's it. Depending on the three settings, *Krdc* works out the other details for the connection. The app's set of choices is rather limited, but you do get the option to manually specify a resolution for the session.

Krdc lists all connected remote computers in different tabs. From within a connection, you get buttons to switch to full screen, scale the remote display to fit the local resolution, take a screenshot of the remote display, change the session into a view-only mode, and send all keyboard inputs to the remote computer.

We connected to another computer on the local LAN using the default medium setting. Videos played flawlessly albeit without sound, and our

"Krdc's main interface might seem overwhelming, but it's fairly simple to operate."



Krdc doesn't offer the option to route the connection through a SSH channel.

test PDFs were readable and scrolled nicely as well. However, performance degraded sharply when we selected the High quality option which is suggested for LAN computers. Inversely, the Low quality wasn't of much use as the colours and fonts rendered poorly.

VERDICT

A useful client for KDE users who use VNC occasionally.

★★★★★

Vinagre

Not nearly as versatile as vinegar.

Vinagre, also known simply as Remote Desktop Viewer, is Gnome's default client for viewing remote desktops and supports the VNC, RDP, SPICE and SSH protocols. It has a minimal interface that's very much like *Remmina*. However, there aren't nearly as many advanced options that are available behind *Remmina*'s simple GUI. To connect all you need to do is pick a protocol from the pull-down list and enter its IP address. There's also a very helpful Find button next to the host address field that hunts for active servers on the local network.

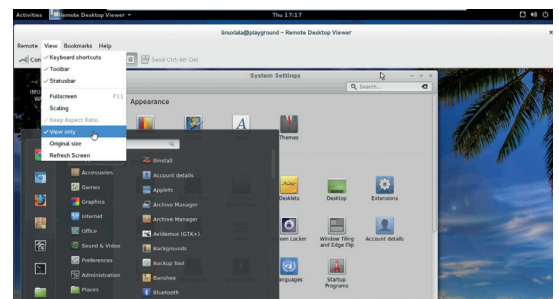
Also much like *Remmina*, you get optional checkboxes for starting a full-screen session, a view-only or a scaled window. You also have the ability to select a colour depth from 24-bit true colour to three-bit ultra-low colour. You can also enable JPEG compression

if you have the resources to bear the processing overhead.

Another useful option is the ability to tunnel the VNC connection through an SSH server. To establish a secure session make sure you run the SSH server on the remote server that's also the VNC host.

Also remember that before switching to the full-screen mode, you should enable the keyboard shortcuts option (under the View menu) and then use the F11 key to switch between fullscreen and window mode. During an active remote desktop session, you can stop sending keyboard and mouse to the remote desktop and effectively turn it into a view-only session. The interface also includes a button to send the famous three-finger salute (Ctrl+Alt+Del) to the remote desktop.

Performance wise, the app is pretty mediocre. The *Snake* game was



The Reverse Connection option simplifies the process of accessing a host behind a firewall.

playable at all quality levels and while videos played without jerks on the 16 bit High Colour setting they weren't really watchable because of the lack of colours. If you went any higher, the videos became jerky and started skipping frames. Also, Vinagre doesn't send audio, and we had to disconnect to change the quality setting.

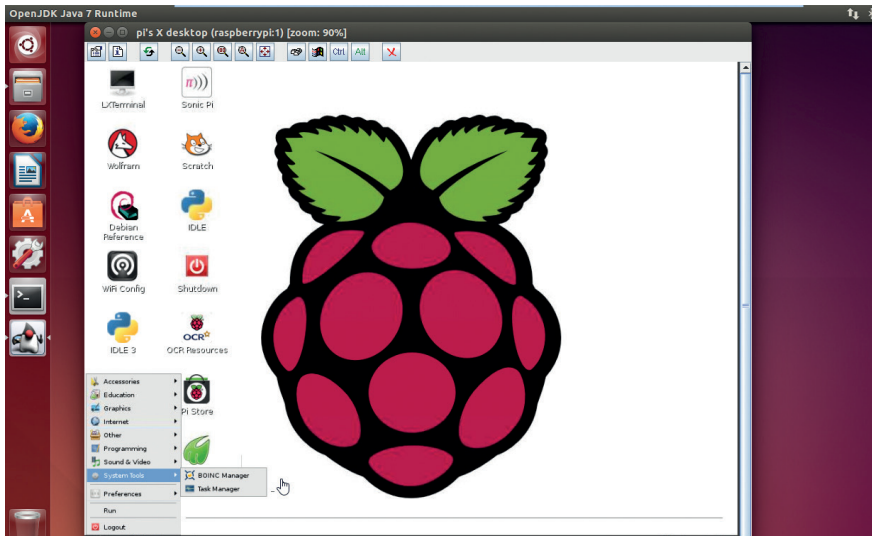
VERDICT

Vinagre is to Gnome what *Krdc* is to KDE – a usable default for occasional use.

★★★★★

TightVNC

Never let go of your computers.



The project has recently released an Android client optimised for mobile internet connections.

TightVNC is one of the oldest VNC client that's still in development and is the progenitor of many popular VNC clients with different goals. The *TightVNC* project doesn't just produce a client. They also release a VNC server, which is what we use on the Raspberry Pi.

The project uses its own enhanced version of VNC's RFB protocol. The project has added extensions to the RFB protocol to improve performance over low bandwidth connections. *TightVNC* gets its name from the fact that it encodes the VNC stream more tightly by using a combination of the JPEG and Zlib compression mechanisms. However, this compression shouldn't impose any performance penalties on modern processors. In fact, the official Raspberry Pi documentation asks users to run the *TightVNC* server to set up a VNC on the Raspberry Pi.

One of the best things about *TightVNC* is that it is still compatible with other implementations of VNC. However to use its tight encoding and gain full advantage of its enhancements you must use *TightVNC* at both end of the connection. The *TightVNC* client for Linux is written in Java that doesn't need to be installed – just double click on the **.jar** file to launch the client (assuming you have installed the JRE). It works fine with the OpenJDK JRE. We tested the *TightVNC* client by connecting it to the *Vino* server as well

as its own *TightVNC* server. When using *Vino*, we had to turn off encryption on the server before *TightVNC* would connect to the server. It also correctly autodetected colour depth. Video playback without the audio was watchable and the Snake game was playable without any issues.

Some fiddling required

When using *TightVNC* server, we had to alter the default config file to show the server's MATE window manager. We had to do a similar modification to view Ubuntu's Unity desktop as well. This connection uses the Tight protocol, although we didn't notice any remarkable improvements. One major difference is that the *TightVNC* server shares a new desktop, while *Vino* shares the same desktop that's currently on the remote desktop.

In addition to the lack of audio from the remote desktop, there's also no means to chat with the user on the other end. Furthermore, the file sharing facility is only supported on the Windows platform. You wouldn't be able to transfer files even if you use the *TightVNC* Linux client to connect to the server on Windows.

VERDICT

A decent performer that covers all your multi-platform multi-device bases.

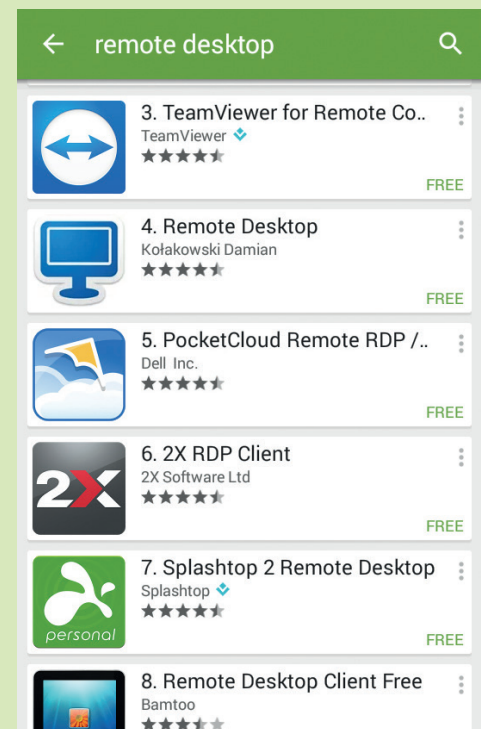
★★★★★

Other options

There are several other remote desktop clients and solutions that you can use. While we have only covered the most popular and actively developed VNC clients, there are several others. There's *RealVNC*, which is often touted as the official VNC client. You can use it for free for a limited number of connections after registering on its website. Other VNC clients include *TigerVNC*, *TurboVNC* and *x11vnc*. If you're using Microsoft's RDP server, you can connect to it using the *Rdesktop* client.

There's also the open source *Neatx* server based on *NoMachine*'s NX technology, and Google's *Chrome Remote Desktop* tool which is currently in beta for Linux. If you aren't averse to proprietary solutions, there are several paid and freeware solutions such as *Bomgar* and *Mikogo*.

One solution that uses remote desktop sharing technology is *iTALC*. It is an open source solution for remotely accessing, controlling and managing classroom computers and uses the RFB protocol. Also, if you just wish to access a single app and care more about security than zippyness then you can just enable X11 forwarding over SSH.



Google Play has several clients for accessing remote desktops on your Android device.

NoMachine NX vs TeamViewer

Cross-platform proprietary freeware at their best.

NoMachine NX uses the NX protocol that tunnels a remote X session across an SSH encrypted channel. The protocol also encodes and compresses data to minimise the bandwidth required. This allows it to do some cool things such as pipe audio from the remote server to the local client.

The tool can automatically pick up any NX servers that are accepting connections running on the LAN. You can also define a new connection by specifying its IP address and the login credentials that you specified while setting up the server. *NoMachine* will then detect the remote resolution and offer to change it to match the local resolution. By default, it'll forward audio to the local client and mute it on the server, but you get the option to unmute it on the server as well.

Once connected you can access all its features from the Session menu, which is accessed from the page peel in the top-right corner of the window. The menu gives you access to some useful features, such as the ability to access a device such as a disk, or printer, stream the mic input to the remote server, and record the remote desktop in a WebM video.

NoMachine lets you export the contents of a local disk to the remote machine or import the remote disk into the local desktop. You can also copy files by dragging them to and

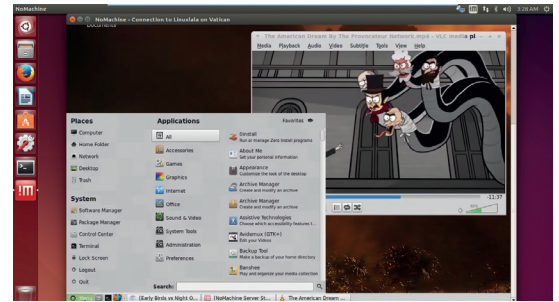
from the remote desktop. You can mount a remote disk either as public, which mounts it in `/media` or private which mounts it on the user's desktop. Similarly, you can also manage local and remote printers and USB devices including removable disks, scanners, web cams and more. Performance-wise *NoMachine* NX is phenomenal. Video playback, games and PDFs look and work as if you are operating them on the local computer. There's no noticeable lag and the images and video are very crisp.

One for the team

TeamViewer is perhaps one of the most recognisable names in the remote desktop domain and the app is used by several major enterprises. However, its Linux client isn't nearly as spectacular. In fact, *TeamViewer's* Linux client still runs with the help of *Wine*, like its initial version several years ago.

The client offers more features than you get with the usual open source remote desktop clients, and *TeamViewer* uses its own proprietary protocol that connects clients through a central server. One obvious advantage of this scheme is that you can connect to a remote desktop from anywhere, even those behind firewalls, without messing with routers or setting up port forwarding.

TeamViewer is one of the most convenient apps to set up and use. Just



NoMachine places an icon in the system tray that gives you access to tools such as the whiteboard, which can be used for scribbling instant messages.

launch the client and enter the unique numeric code displayed on the machine you wish to connect to. If you're accessing your own remote computer, you can set up a password and log in unattended. If you are handholding another *TeamViewer* user, all you need from them is their unique code and the randomly generated password that'll be valid only for the current session.

However, the quality of the connection is very poor. You can use it to either render a nice desktop at

“Video playback, games and PDFs look and work as if they are on the local computer.”

a slow frame rate or an unreadable desktop at a usable speed. For what it's worth, *TeamViewer* does offer a few extra features such as a text and video chat client, the ability to transfer files as well as a VoIP service. The client can also host group meetings. Some features though, such as the ability to invite other users into a session, require you to sign into a *TeamViewer* account. Like *NoMachine*, you can also record a session in *TeamViewer's* own **.tvs** format. But we couldn't get the option to convert it into AVI format even after following the manual.

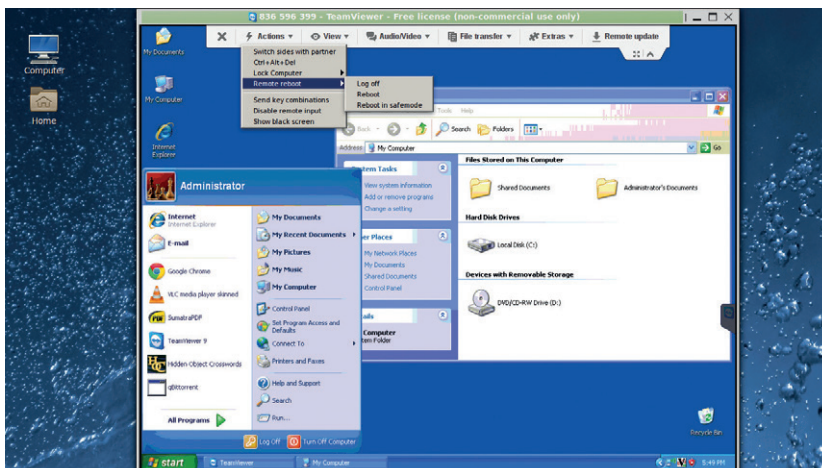
VERDICT

NOMACHINE NX The best remote desktop solution for the pragmatic user.

★★★★★

TEAMVIEWER It might be a big name on Windows but on Linux it's just a bit meh.

★★★★★



Both *NoMachine* and *TeamViewer* can be installed on any Linux distribution, and have precompiled binaries for RPM- and Deb-based distros.

OUR VERDICT

Remote desktop clients

KDE's *Krdc* and Gnome's *Vinagre* are good defaults for their respective desktops and both don't support NX connections. One big turn off with *Krdc* is the limited flexibility with the quality settings which really just left us with only one usable quality option. *Vinagre* didn't fare much better with its mediocre performance.

TightVNC is the only Java app on test. The app performed well even when used with other VNC servers. One big advantage of the app is that it's cross-platform and even has mobile clients. It's


change the quality settings of the connection on the fly. All things considered, *Remmina* is a wonderful remote desktop client and would serve you well if you can live without extra features such as the ability to transfer files

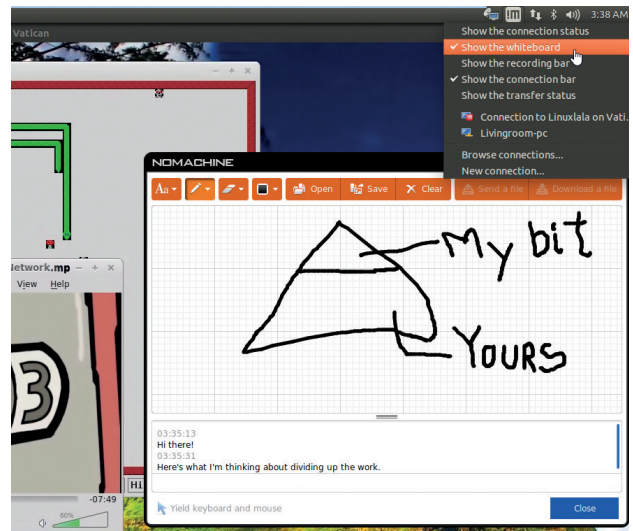
That leaves us with the two proprietary clients. *TeamViewer* turned out to be a disappointment. Its ease of use doesn't outweigh its poor performance and we can't recommend it for any use case. *NoMachine NX* surprised us too, pleasantly, that is. We haven't yet awarded a Linux Voice Group Test to a proprietary app, but *NoMachine*

"It's proprietary, but NoMachine NX is way ahead of everything out there."

the recommended server for the Raspberry Pi, and if you're using it on the Pi you should use the client to take advantage of the protocol's tight compression.

However, for maximum coverage there's no beating *Remmina*. The client supports the widest range of protocols and will connect to all kinds of remote desktop servers. The app scores well in the performance department as well and gives you the flexibility to

NX is way ahead of everything out there. The open source clients do their bit too and depending on your use case, might be the perfect tool for many of you. However they aren't as comprehensive as *NoMachine NX*, which will work for all types of use cases. You can use it to simply access your remote desktop or any peripherals attached to it or use it to collaborate with another remote user over the internet without much fuss. 



At least the NX protocol is open.

1st NoMachine NX 4.3.30

Licence Freeware Version 4.3.30

www.nomachine.com

Proprietary software done right. The quality is so good, we have to recommend it even though it isn't free software.

2nd Remmina 1.1.1

Licence GNU GPL Version 1.1.1

<http://freedp.github.io/Remmina>

Supports the widest range of protocols and performs well.

3rd TightVNC 2.7.10

Licence GNU GPL Version 2.7.10

www.tightvnc.com

Best coupled with its own server that's tuned to make best use of limited resources.

4th Krdc 4.14.1

Licence GNU GPL Version 4.14.1

www.kde.org/applications/internet/krdc

Good default client for simple use.

5th Vinagre 3.12.2

Licence GNU GPL Version 3.12.2

<https://wiki.gnome.org/Apps/Vinagre>

Another good default for occasional use.

6th TeamViewer 10

Licence Freeware Version 10.0

www.teamviewer.com

Overshadowed by every client in terms of performance.

	VNC	RDP	NX	File Transfer	Audio Support
Remmina	Y	Y	Y	N	N
Krdc	Y	Y	N	N	N
Vinagre	Y	Y	N	N	N
TightVNC	Y	N	N	N	N
NoMachine NX	N	N	Y	Y	Y
TeamViewer	N	N	N	Y	N