

# Bachelor Degree Project



UNIVERSITY  
OF SKÖVDE

## **Remote desktop protocols** A comparison of Spice, NX and VNC

Bachelor Degree Project in Network and  
System administration  
G2E 15h  
Spring term 2012

Martin Hagström  
h06marha@student.his.se

June 5, 2012

Supervisor: Jonas Mellin  
Examiner: Thomas Fischer

## **Remote desktop protocols**

This report is submitted in partial fulfillment of the requirements for the Bachelor's degree in Computer Science at the Institution for Communication and Information. This project has been supervised by: Jonas Mellin

**June 5, 2012**

All material in this report which is not my own work has been identified and no material is included for which a degree has previously been conferred.

Signed: \_\_\_\_\_

## **Abstract**

This thesis compares the remote desktop protocol Spice to NX and VNC taking into consideration user experience when viewing multimedia content. By measuring the quality of the protocols by viewing a video in a slow-motion benchmark compared to ordinary speeds it is shown that Spice has a low video quality compared to VNC. It is likely that due to a large amount of data sent, Spice does not manage to reach a high quality user experience.

**Keywords:** Spice, NX, VNC, remote desktop protocol, user experience, video quality

# Contents

- 1 Summary** **1**
- 2 Introduction** **2**
- 3 Background** **3**
  - 3.1 Use of remote desktop protocols 3
    - 3.1.1 Telecommuting 3
    - 3.1.2 Virtual desktop environment 3
    - 3.1.3 IT support 4
  - 3.2 User experience 4
  - 3.3 Types of remote desktop protocols 4
    - 3.3.1 Pixel-based remote desktop protocol 4
    - 3.3.2 Object-based remote desktop protocol 5
    - 3.3.3 Comparison of pixel-based and object-based remote desktop protocols 5
  - 3.4 Examples of remote desktop protocols 5
    - 3.4.1 Spice 5
    - 3.4.2 NX 7
    - 3.4.3 VNC 7
    - 3.4.4 Comparison 7
    - 3.4.5 Previous comparisons 8
- 4 Problem** **9**
  - 4.1 Purpose 9
  - 4.2 Motivation 9
  - 4.3 Partial Goals 9
- 5 Method** **10**
  - 5.1 Experimental approach 10
  - 5.2 Slow-motion benchmarking 10
  - 5.3 Environment 11
  - 5.4 Setup of environment 11
    - 5.4.1 Server configuration 11
    - 5.4.2 Client configuration 13
    - 5.4.3 Load generator configuration 13
    - 5.4.4 Network configuration 14
    - 5.4.5 Encoding of videos 14
- 6 Result** **15**
  - 6.1 Low bandwidth 15
    - 6.1.1 Video quality 15
    - 6.1.2 Data usage 16
  - 6.2 Medium bandwidth 16
    - 6.2.1 Video quality 17
    - 6.2.2 Data usage 17

<b>7</b>	<b>Analysis</b>	<b>18</b>
7.1	Spice . . . . .	18
7.2	NX . . . . .	18
7.3	VNC . . . . .	18
7.4	Related work . . . . .	18
<b>8</b>	<b>Conclusion</b>	<b>19</b>
8.1	Discussion . . . . .	19
8.1.1	Ethical aspects . . . . .	19
8.1.2	Social aspects . . . . .	20
8.2	Future work . . . . .	20
8.2.1	Web application . . . . .	20
8.2.2	Voice over IP quality measurements . . . . .	20
8.2.3	Different configurations . . . . .	20
	<b>References</b>	<b>21</b>

# 1 Summary

This work examines the remote desktop protocol Spice compared to NX and VNC. A remote desktop protocol is computer software that allows a user to control a computer remotely. There are two ways this can be done. Either by copying what is shown on the remote computer to the local computer or by copying the instructions that draw the screen. The advantage of copying instructions instead of the entire screen is that it can be more efficient as less data is sent over the network. The downside is that these instructions can be complex which requires a more powerful local computer.

Since multimedia is becoming a greater part of everyday computer use, this report examines how the remote desktop protocols handle playing video clips. The creators of Spice claim that they can achieve a high quality user experience but the protocol has not been the focus of any prior study. Therefore it is examined against NX and VNC which have both been frequent in other studies.

The protocols are examined by using slow-motion benchmarking. By playing a video in slow-motion, all of the video data is transferred. When played in ordinary speeds some of the data might get lost. By comparing the data sent in slow-motion to the ordinary speed, a video quality percentage is found.

The result from the experiments show that Spice sends a lot of data, between 5 and 80 times the size of the source video. Because of this, it has poor video quality. NX does not send as much data as Spice but still has similarly poor video quality whereas VNC has the best quality, about 4 times as good as NX and Spice. However, this type of measurement might not give an accurate indication of quality. The poor quality achieved might still be good enough when seen by a user. Because of those deficits, for future work it is suggested videos are shown to the users to give a score on the quality. This could be compared to the results discussed in this work to see if there is a relation between the results.

## 2 Introduction

The use of remote desktop protocols is common in current work environments. These protocols allow users to access remote computers as if they were local desktop computers. This allows the use of thin clients in virtual desktop environments, telecommuting and doing support and administering hosts remotely. However, since the data being sent is often compressed or sent over a poor connection, the user will not get the experience of a desktop computer. Often there is latency in graphical updates or mouse movement. This is an issue as your work is constantly being delayed or interrupted making it an annoying experience.

The purpose of this work is to analyze critical features of the remote desktop protocol Spice compared to NX and VNC. It will give an indication of how these protocols behave in a typical environment. The attributes that will be tested are multimedia performance and if there is time, web application performance. The expected result is that Spice provides a better user experience than NX and VNC when working with multimedia applications.

As different remote desktop protocols focus on different features such as multimedia performance or operating system independence, it is important to know which one is most suited for the environment where it will be implemented. Therefore an objective comparison needs to be made to give a system administrator the tools needed to make a good choice.

The main scope of this report is to compare three selected remote desktop protocols. Focus is put on the end users experience. Therefore, the comparisons made will not take other characteristics such as different network setups into account. The environment is limited to that offered by lab environment of the University of Skövde.

## 3 Background

This section describes in what environments remote desktop protocols are used, what different types of remote desktop protocols there are and gives details of the three protocols that will be compared.

### 3.1 Use of remote desktop protocols

There are many reasons to use remote desktop protocols. They can, for example, be used when telecommuting to access office resources. They can also be used in virtual desktop environments, typically with thin clients. They are also common to use when providing support.

#### 3.1.1 Telecommuting

Telecommuting is the act of working remotely with the help of information technology. A key component in telecommuting is the use of remote desktop protocols which gives access to work resources from anywhere. According to Sikes, Mason, and VonLehmden (2011), telecommuting can bring many advantages both to a company and to the individuals. They state that the company can save space and money by not having to pay for office space. In addition, by allowing flexible work hours employee motivation increases. Further, it can also give benefits for the environment by decreasing the amount of travels needed. From the individual's perspective, Harpaz (2002) identify many advantages. They state that when working independently the individual's control over work occurs more freely and naturally. In addition, it gives the individual flexibility when planning work hours which allows other tasks to be balanced such as activities involving family and community. There is also less time being spent on distractions that often occur in a cubicle environment.

#### 3.1.2 Virtual desktop environment

Remote desktop protocols can be used in virtual desktop environments. The remote desktop protocols provide a link between a thin client or desktop computer to network resources such as virtual devices with other operating systems or development and testing environments.

A typical use of a virtual desktop environment consists of a server virtualizing host machines. These host machines are accessed with the help of a connection broker that handles how connections are made to the hosts. The connection broker can monitor a pool of virtual hosts and provide load balancing for new connections.

The use of virtualization gives many benefits (Miller & Pegah, 2007). It consolidates the use of machines which makes it easier to use more of their performance. Virtualization also makes administration easier, upgrades and installations can be done on a single machine which is then propagated to all the virtual hosts.

A virtual desktop environment is typically used together with thin clients. They are cheap devices that provide an interface between the remote user and the server. The thin client has connections for mouse, keyboard, monitor, network etc. Together with a remote desktop protocol they can act as a normal desktop computer at a low cost and with no moving parts, making it easier to support.



### **3.1.3 IT support**

Remote desktop protocols can be used when doing IT support (Baratto, Kim, & Nieh, 2005). By allowing both the user and the support personnel to control the computer at the same time, support can become more effective. It can also make it easier as you can control the computer yourself instead of having to explain the procedures to someone who might have much less experience. In a large office environment, not having to go to the physical computer speeds things up, increasing productivity.

## **3.2 User experience**

Lubonski, Gay, and Simmonds (2005) state that remote desktop protocols were designed and optimized to work with data-oriented programs in a LAN environment over low-bandwidth links. They also state that even though remote desktop protocols work efficiently in this environment, they perform poorly in office environments with multimedia applications such as audio and video-conferencing, multimedia presentations and e-learning. They argue that this lack of performance significantly decreases end-user satisfaction.

An example of when user-perceived performance deteriorates is when the number of displayed frames per second in a video clip is reduced below a threshold (Lubonski et al., 2005). Video becomes an issue as it has real-time properties such as the need for audio and video to be synced. It also becomes difficult for some protocols because video is difficult to objectify.

While some features such as video performance are easy to measure, other features are more subjective such as how fast a menu opens or if there is lag when scrolling a long web page. Because screen updates are very frequent when playing a video, the remote desktop server has to encode a lot of frames, making it a good indicator of performance (Deboosere et al., 2007). When protocols are designed, compromises are made. Because of this, different protocols may handle video differently, making it an interesting topic to experiment on.

## **3.3 Types of remote desktop protocols**

All remote desktop protocols provide the user with a link to the remote host. Input from the keyboard and mouse is sent over the network to the remote computer. The remote computer interprets the instructions and sends back the result in the form of an image. In this way, the user controls the remote computer as if using it locally.

According to Baratto et al. (2005) there are two major subtypes of remote desktop protocols. They state that the first one uses raw pixel data when sending back the image while the other one uses drawing primitives named object-based remote desktop protocols.

### **3.3.1 Pixel-based remote desktop protocol**

Using raw pixel data gives a great flexibility at the cost of performance. These types of remote desktop protocols work by sending a copy of the frame buffer, which keeps a record of all the pixels that are displayed on the screen, to the remote desktop client. This allows pixel-based remote desktop protocols to work with any type of windowing

system, operating system or application (Richardson, Stafford-Fraser, Wood, & Hopper, 1998).

A common remote desktop protocol that uses this is the remote frame buffer protocol (RFB). It is the protocol used by Virtual Network Computing (VNC) which is a widely-used desktop sharing system (Baratto et al., 2005) with many variants that work on most operating systems.

### **3.3.2 Object-based remote desktop protocol**

Remote desktop protocols can draw the image on the local host by using application-level display commands sent by the remote server, for example, drawing boxes and writing text. These commands are interpreted by the local host and drawn on the screen. Because of this, the client machine requires more computing power than when using the raw pixel type of remote desktop protocol.

An example of this type of remote desktop protocol is the X window system which can forward application-level display commands over a network. This allows the local client to process user interface actions that do not involve application logic without having to send any data over the network. However, since user interface and application logic is often tightly coupled, there is a need for synchronization which can give a poor user experience.

There have been extensions of the X system such as low-bandwidth X (LBX) and NX that address some of the issues of X.

### **3.3.3 Comparison of pixel-based and object-based remote desktop protocols**

When dealing with units with limited resources such as smartphones, pixel-based protocols can be easier to implement since they only need to handle raw bitmaps being sent. The downside to this is that sending large amounts of data can be expensive in terms of bandwidth compared to object-based protocols. Even with the use of compression, noticeable lag often appears on these types of remote desktop protocols, especially if the connection is poor. A object-based protocol on the other hand uses the client window system. This can become a more complex environment to maintain as they can be large systems with many dependencies. As applications become more complex, the local window system needs to be able to keep up making hardware and software replacements likely.

## **3.4 Examples of remote desktop protocols**

There are many different remote desktop protocols. This report focuses on Spice, NX and VNC.

### **3.4.1 Spice**

Simple Protocol for Independent Computing Environments (Spice) is a remote desktop protocol made for the purpose of providing remote computing in a virtual environment. It uses the object-based method of transmitting images to clients.

The rest of this subsection is a summary of the official documentation of Spice (Red Hat, Inc, 2009).

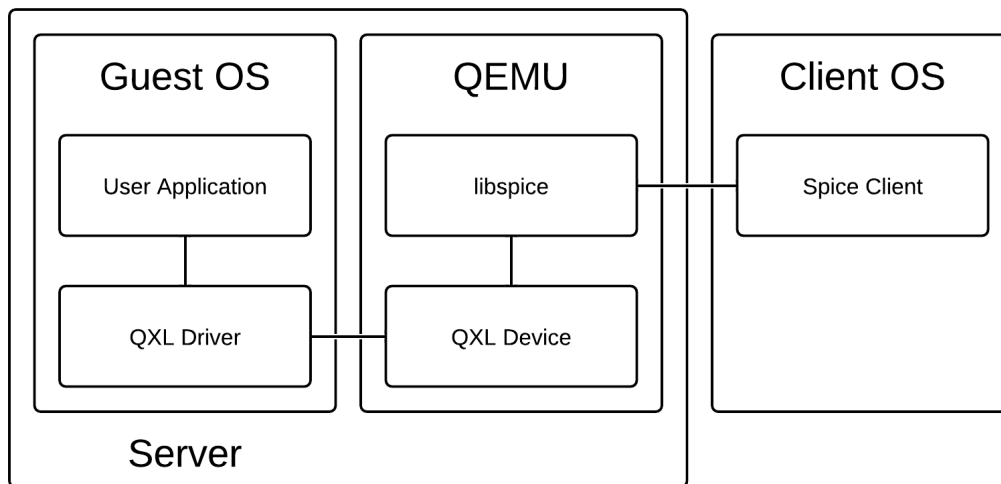


Figure 1: Command flow of Spice protocol. Based on Red Hat, Inc (2009)

Spice consists of three separate parts (see fig 1). A QXL<sup>1</sup> driver on the guest operating system, a server and a client on the client operating system. The QXL driver on the guest operating system is used to increase performance and enhance the graphic capabilities of the guest graphic system. If no drivers exist, standard video graphics array (VGA) is supported.

The server listens for client connections, accepts and communicates with them. The communication is split up in channels with each channel being dedicated to a certain type of data. These channels are: Main, Inputs, Display, Cursor, Playback, Record. Each channel uses a dedicated TCP socket and can be secured with SSL. The server is responsible for:

- Handling channels
- Adding and removing virtual desktop infrastructure (VDI) interfaces (keyboard, mouse, etc)
- Migration process coordination
- Handling user commands
- Communication with guest agent
- Statistics

Spice attempts to pass all rendering tasks to the client to utilize its hardware acceleration abilities. As a last resort, rendering can be done on the server by software or the graphics processing unit (GPU). Graphic commands issued by user applications are sent to the QXL driver. This driver translates the operating system commands to

<sup>1</sup>No explanation has been found what this stands for

QXL commands and pushes them to the QXL device on the server. Libspice then pulls commands from the QXL device to be optimized, for example, if a command is hidden by another command it is dropped. Executing the commands contained in libspice will reproduce the display content from the guest OS. Libspice has other features such as detecting when a video stream is present and maintaining a queue of commands to be sent to the client.

### 3.4.2 NX

NX builds on the X window system. With the help of compression, reduced network round-trips and bandwidth adaptability it improves the usability of X. In addition to this, NX provides session support and user account management which makes it more suitable in large environments with many users (NoMachine, 2012).

Drawing requests sent to NX are stored in a cache with an MD5 checksum associated with it. This allows for identical requests to quickly be fetched from the cache instead of having to send them over the network again since both the server and client keeps a copy of this cache. Only the server needs to calculate and maintain the checksums leaving less work for the client.

Because both the server and client have the exact same cache, the server knows at all times what is available at the client without having to send packets to ask. This is one of the ways NX removes round-trips.

NX uses several different types of compression to increase the user experience on low bandwidth networks. Each different request and reply has default encodings associated with them. In addition, implicit information and padding bits are stripped to lower the amount of data that needs to be transferred.

### 3.4.3 VNC

Virtual Network Computing (VNC) is a pixel-based desktop sharing system. It uses the remote framebuffer (RFB) protocol to transmit its images. VNC uses a server-client model to transmit the framebuffer to the remote client. As this uses a lot of bandwidth several enhancements have been made. For example different encodings and compression (Richardson et al., 1998). Examples of these encodings are:

- Raw encoding — Prints pixels in a left-to-right scanline order.
- Copy-rectangle encoding — A copy is made from one part of the image to another. Typically used when moving a window.
- JPEG encoding — The frame buffer is encoded using joint photo experts group (JPEG) technology.

### 3.4.4 Comparison

As can be seen in table 1 all of the protocols are available in a Linux environment while some have Windows and Mac versions too. NX have some variations such as FreeNX, NoMachine NX and QVD while Spice has none. VNC on the other hand have many, each with different features added which makes it difficult to compare VNC as a whole to other protocols.

	Spice	NX	VNC
Type	Object-based	Object-based	Pixel-based
Availability	Free	Free and enterprise	Free and enterprise
Variations	None	Some	Many
Server software	Linux	Linux	Windows, Linux, Mac
Client software	Windows, Linux	Windows, Linux, Mac	Windows, Linux, Mac

Table 1: Comparison of the remote desktop protocols

### 3.4.5 Previous comparisons

NX and VNC have been part of previous comparisons, for example, by Baratto et al. (2005). By measuring performance on common web and multimedia applications in various network setups they found that due to its aggressive compression techniques NX performed well in normal web applications. In multimedia applications however it performed poorly. Similarly, VNC was found to perform well in web applications while not reaching high quality in multimedia applications.

As Spice has not been part of any comparisons it would be interesting to see if it can match or even outperform NX and VNC.

## 4 Problem

This section describes the purpose, motivation, partial goals and approach of the report.

### 4.1 Purpose

The purpose of this report is to compare and contrast critical features of the remote desktop protocol Spice compared to NX and VNC. Spice is a fairly new protocol that claims to be able to deliver a high quality user experience. This report will give an indication if the creators of Spice are correct.

### 4.2 Motivation

Due to that remote desktop protocols are tightly coupled with the use of thin clients and virtual environments, they are a major part of how users interact with their computers in these environments. To give the users the best possible experience, it is important to know what features are important to consider in a remote desktop protocol and how they perform. As there have been no objective comparisons of the Spice protocol, it is interesting to investigate if the user experience they claim to support lives up to their promises.

Even though NX has performed poorly in previous multimedia comparisons, new versions of the protocol have been released that are said to improve video quality (NoMachine, 2010).

It is also important to note that although there is a free version of NoMachine NX available, it is limited to 2 concurrent connections and 2 users. In an enterprise environment, a more suitable version which has no limit on concurrent connections or users costs a lot of money. As Spice is open source it is important to see if it has equal performance and as such might be an alternative. Compared to VNC, Spice is objects-based instead of pixel-based which should give improvements in efficiency.

### 4.3 Partial Goals

The work has been divided in two parts. The first goal is to set up an experimental environment to give indications on how the two protocols perform when handling multimedia applications. If there is enough time, the second goal is to evaluate web application performance.

## 5 Method

In order to evaluate the remote desktop protocols, an experimental environment is set up in the lab environment of the University of Skövde. The experiment performed measures the quality of the protocols with the help of slow-motion benchmarking.

### 5.1 Experimental approach

An experimental approach is chosen because it mimics the approach done by Baratto et al. (2005) and Deboosere et al. (2007). An experimental approach is needed as it is difficult to judge user experience with only analysis. It also makes it possible to compare the results with previous work which would not have been possible with only analysis. The environment will contain a server running virtualization software and the protocols that are to be evaluated. There are many different NX implementations, FreeNX is used as they provide a free version of NX that is available in the Fedora repository. Client software for the remote desktop protocols is set up on a client computer. Video clips with varying bitrates are then played on the guest OS through the remote desktop client. The quality of the videos are measured with slow-motion benchmarking.

### 5.2 Slow-motion benchmarking

The method used to evaluate the quality of video transferred through a remote desktop protocol is slow-motion benchmarking as explained by Nieh, Yang, and Novik (2003). By running this slow-motion benchmarking a quality percentage is received. This number gives an indication on how well the protocols perform when displaying multimedia content. It is chosen because it was used by Baratto et al. (2005) in their experiments.

Other methods could be chosen instead of slow-motion benchmarking, for example, letting people watch video-clips and grade them similar to mean opinion score (MOS) that is used when grading telephone call quality. Slow-motion benchmarking is chosen instead of this because of its more objective measurement of quality. A large amount of people would have to view the video clip to be sure that the grade it is given is correct. This becomes difficult to manage when there are many tests to run.

The problem with running benchmarking without a slow-motion baseline is that it is difficult to be certain that the content generated on the server is being sent to the client without the server discarding some information due to, for example, congestion. By running the experiment in slow-motion, enough time is given to allow all the information to be transferred. This can then be compared to the experiment when run in ordinary speeds which provides a percentage of how well the protocols performs compared to ideal conditions. The equation below describes the relation between a baseline and a slow-motion video transfer.

$$VQ(P) = \frac{\frac{Data\ Transferred(P)/Playback\ Time(P)}{Ideal\ FPS(P)}}{\frac{Data\ Transferred(slow-mo)/Playback\ Time(slow-mo)}{Ideal\ FPS(slow-mo)}}$$

For example, a 30 second long video transferred in slow-motion in 12 minutes at 1 frame per second (fps) with 50 MB of data compared to 1 minute at 24 fps and 50 MB

would produce these numbers:

$$VQ = \frac{\frac{50/60}{24}}{\frac{50/720}{1}} = 50\%$$

Because the clip takes twice the time it should take to complete, the video quality is 50%. If the data transferred was reduced to 25 MB to make the clip play at the correct speed the video quality would still be 50% because of the loss of data:

$$VQ = \frac{\frac{25/30}{24}}{\frac{50/720}{1}} = 50\%$$

It should be noted that even though this number is low, it might be acceptable for a user as it can be difficult to spot a difference in quality.

### 5.3 Environment

In the general use of remote desktop protocols, a server is connected to a client over some sort of network, see fig 2. From this model, several variables are available for experimentation. Difference in bandwidth is common even when working in a LAN environment. With the use of a router, arbitrary speeds can be set on different links. It is also likely that other services are run on the same network. This can be emulated with the help of load generators. Another variable is the difference in bitrate in videos. A video conference call require much less bitrate than a high quality movie.

There are other variables that are not included in this experiment but should still be noted. CPU load is one of these. Even though CPU load might affect the remote desktop protocols, the focus lies on the performance of the protocols over the network and not on the computers performance. There are also many different network properties that could be examined, for example round-trip time, jitter and packet size. Even though these properties are important they are deemed as less important than the chosen properties. It is possible that these variables vary during the test, this is handled through the help of descriptive statistics by running the tests multiple times to make sure no large deviations exist. In future work, all of these variables could be taken into account.

In order to emulate the variables discussed above, the units seen in fig 3 are used in the experiment.

### 5.4 Setup of environment

The experimental environment is set up in the lab environments of the University of Skövde. The configurations used are described below.

#### 5.4.1 Server configuration

The server is a HP Compaq dc7900 with an Intel Core 2 Quad Q9400 2.66GHz processor and 8GB memory. It is installed with Fedora 16 x86-64 (Linux 3.3.0) running Gnome 3.2.1. Since spice is run on the virtualization server and not the host, spice is installed on the server. In order to provide virtualization, Quick Emulator (QEMU) is installed as this was recommended by spice. An image is created to hold the virtual hard drive for the virtual machine with the following command:



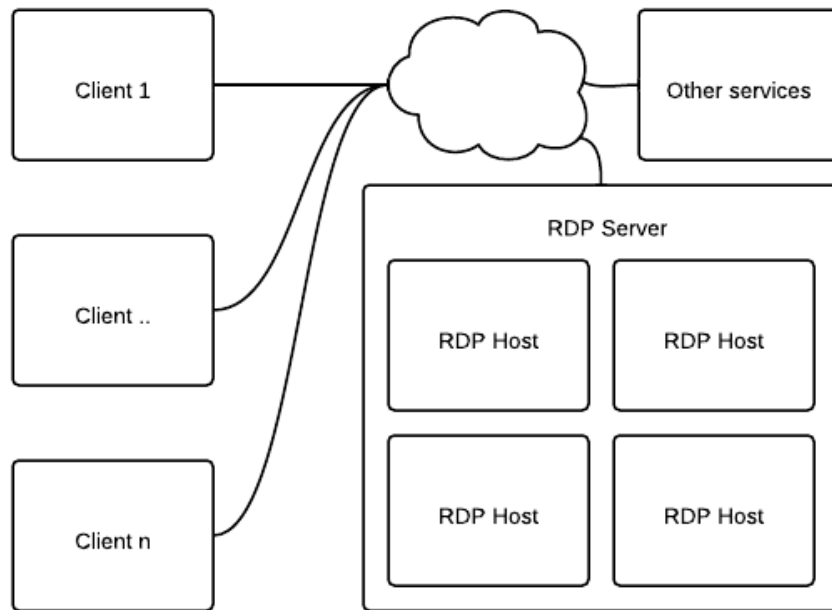


Figure 2: General remote desktop protocol network

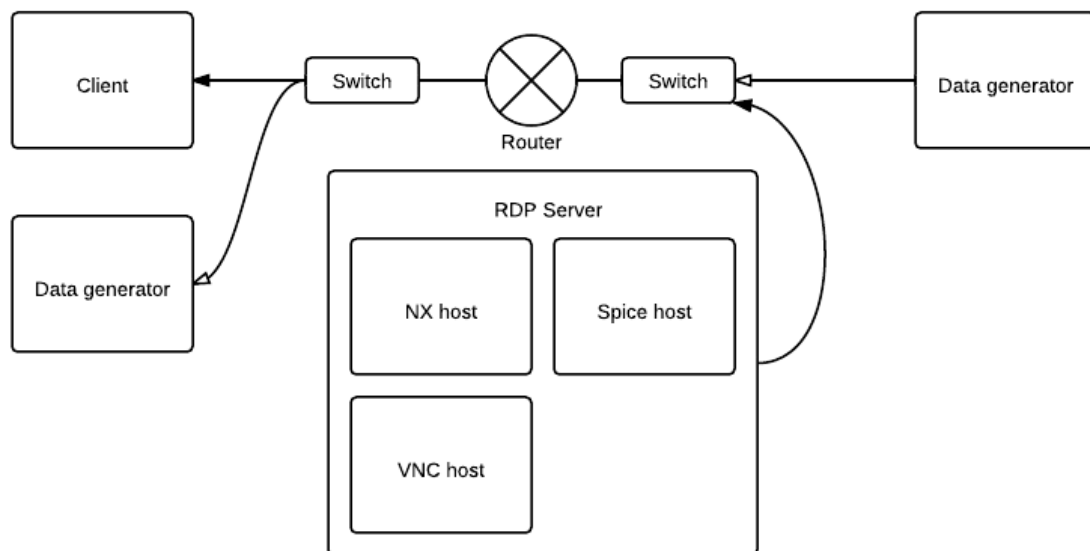


Figure 3: Emulated remote desktop protocol network

```
qemu-img create spice.img 10G
```

After installing the necessary basic software on this image, it is copied to two new images. This is done in order to provide exact copies of the setup and configuration of the virtual hosts. The only additions that are made on these machines is installation of the respective remote desktop protocol server software. All virtual machines uses MPlayer SVN-r34129-4.6.2 to play the video clips. For the slow-motion benchmarking, the following command is used to play the video:

```
mplayer -fs -fps 2 bird06.avi
```

When playing the video in ordinary speed, the following command is used:

```
mplayer -fs bird06.avi
```

### 5.4.2 Client configuration

A client machine is set up on a HP Compaq dc7600 with an Intel Pentium D 2.8GHz processor and 4GB memory. It is installed with Fedora 16 x86-64 (Linux 3.3.0) running Gnome 3.2.1. The client software for the remote desktop protocols are installed with the command

```
yum install spice-client freenx-client tigervnc
```

In the client software for NX and VNC, there are settings that might have implications on the quality. For NX, the estimated bandwidth can be selected. In this software modem, integrated services digital network (ISDN), asymmetric digital subscriber line (ADSL), wide area network (WAN) and local area network (LAN) are selectable. For VNC, different encodings can be selected manually or automatically. Custom compression levels can also be selected. To keep the comparison as fair as possible, since Spice offers none of these settings, automatic or suggested values are used where possible. The only choice that has no automatic setting is NX's bandwidth setting. It is put to WAN for the low bandwidth tests and LAN for the medium bandwidth test. This is done because spice automatically detects how much bandwidth is available and would otherwise give an unfair advantage to spice. All tests are run in the resolution 1280x1024.

### 5.4.3 Load generator configuration

Two machines are set up to be used as load generators. They are HP Compaq dc5750 with an AMD Athlon 64 X2 Dual Core 3800 processor and 2GB memory. A liveCD with Ubuntu 10.04 is used together with Iperf. Iperf is a program that generates load by sending data between a server and a client. It can be configured to use TCP or UDP, bandwidth to use, TCP window size, TCP maximum segment size and more. For this experiment, default settings are used which makes Iperf send as much data as the network allows while the settings are negotiated for optimal performance.

One machine is set up as the server while the other is used as a client. The direction of the transmission is the same as the direction of the video transfer of the remote desktop protocol in order to make sure that the remote desktop protocols are affected by the network load.

#### 5.4.4 Network configuration

A Cisco 2901 router with IOS 15.1 is used for the connection between servers and clients. It is set up to provide dynamic addresses to the other machines on the network. With the help of the **speed** command on the interfaces, different speeds can be set.

The two switches used are Cisco 2960's with IOS 12.2. They are needed because of the lack of ports on the router. Due to the switches only having 100 Mbit ports, the experiment is limited to this speed instead of 1Gbit which was the original intent.

#### 5.4.5 Encoding of videos

A suitable set of videos that could be used in the experiment was not found. Therefore custom videos are made to match the bitrates needed. A 548 frame, 23 second long video that is encoded in 90 Mbit/s bitrate is used as the source material. With the help of mencoder the video is converted to 10 Mbit/s and 0.6 Mbit/s with the following commands.

```
mencoder bird90.mkv -ovc xvid -oac mp3lame -xvidencopts fixed_quant=2 -o  
bird10.avi  
mencoder bird90.mkv -ovc xvid -oac mp3lame -xvidencopts fixed_quant=31 -o  
bird06.avi
```

The size of the video clips are 242 MB and 15MB, other than this and the bitrate they are identical.

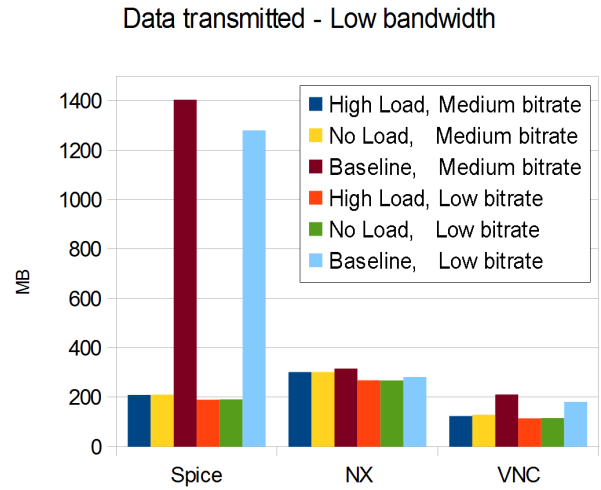
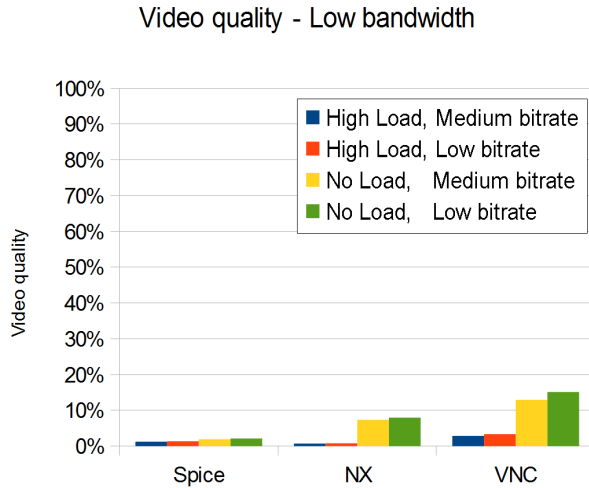


Figure 4: Low bandwidth video quality      Figure 5: Low bandwidth data transferred

## 6 Result

The result of the experiment shows that Spice is the worst performing protocol with low video quality and high data usage. NX is stable with low variations while VNC has both the highest video quality and lowest data transmission.

All but two trials are completed ten times to make sure no large deviations exist. The two test that are not completed ten times had to be cut down to three time due to lack of time. Due to the fact that the data acquired in the experiment did not vary more than 1%, average values are presented in this paper.

### 6.1 Low bandwidth

In a 10 Mbit/s environment, the results show that no protocol has a high video quality. While Spice performs worst, VNC is the only protocol to reach over 10% quality which gives it the highest video quality of the three protocols.

#### 6.1.1 Video quality

Figure 4 shows the result for the protocols when used in a low bandwidth environment of 10Mbit/s. It should be noted that due to time constraints only three tests could be completed of NX when subjected to high loads as each trial took over 45 minutes to complete. Since the other test results are very consistent this suggests that the results still have some credit.

Spice, NX and VNC achieved a video quality of 1%, 1% and 3% respectively under high load playing a medium bitrate video. When lowering the bitrate the quality did not change.

When removing the load NX and VNC showed improvement while Spice only gave very slightly better quality. 2%, 7% and 13% respectively for Spice, NX and VNC during no added load playing a medium bitrate video. When lowering the bitrate, only a slight increase in quality is shown for NX and VNC. 2%, 8% and 15%.

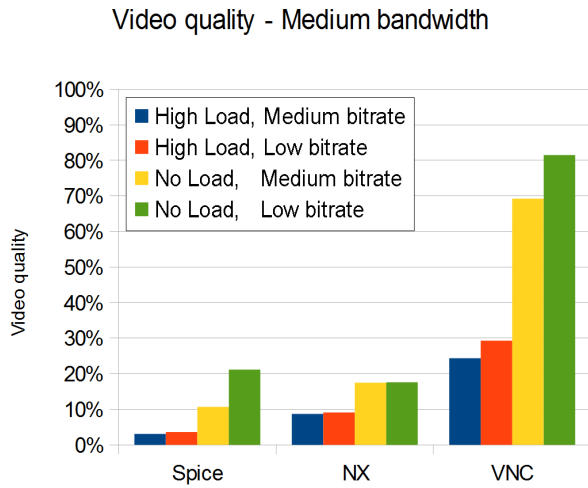


Figure 6: Medium bandwidth video quality

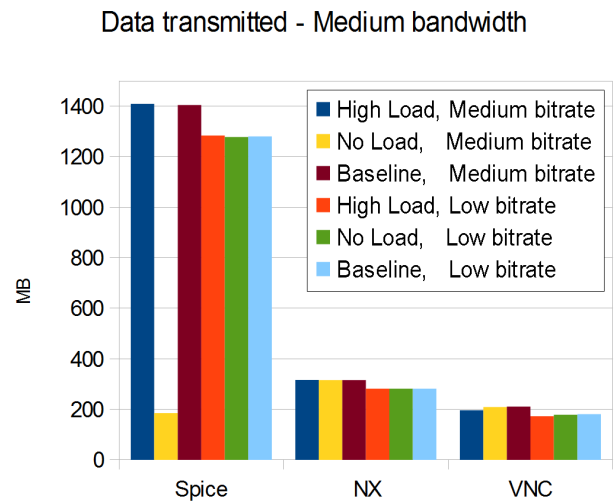


Figure 7: Medium bandwidth data transferred

### 6.1.2 Data usage

It can be seen in figure 5 that the data transferred in the baseline for Spice is very large compared to the source material. At 1404 MB transferred compared to 242 MB of the medium bitrate video, more than 5 times the data of the video clip is sent to the client. When lowering the bitrate, the ratio increases to 80 times the data transferred. 1280 MB compared to 16 MB of the source video. This is not unique to Spice however. The decrease in data transferred is low both for NX and VNC. NX sends 316 MB and 282 MB of data for medium and low bitrates respectively while VNC sends 211 MB and 181 MB.

Spice sends a very low amount of data when viewing the videos in ordinary speed compared to the baseline. The high and no added loads does not give much difference in data sent, 210 MB is sent for the high bitrate video while 190–191 MB is sent for the low bitrate video. The same can be seen when viewing the numbers for NX and VNC, a very small difference is shown for high or no added load. NX sends 302 MB of data for the medium bitrate video and 268 MB for the low bitrate video while VNC uses 124–129 MB and 114–116 MB respectively.

Comparing the data transmitted while watching the video in ordinary speed to the baseline using Spice, it can be seen that a large amount of data is lost. The baseline sends almost 7 times the amount of data as the ordinary speed transmission. For NX the baseline is closer to the ordinary speed transmissions, only 5% less data is transmitted. VNC on the other hand drops around 60% of the data sent.

## 6.2 Medium bandwidth

The results for the 100 Mbit/s environment show that even though the protocols perform better, only VNC reaches above 50% video quality.

### 6.2.1 Video quality

As can be seen in figure 6, when the bandwidth increases so does the quality. At 100Mbit/s with high load and medium bitrate Spice reaches a video quality of 3% with NX and VNC at 9% and 24% respectively. When the bitrate is reduced, the quality for Spice increases slightly at 4% while NX stays at 9%. VNC's increase is a bit bigger at 29%.

Without added load on the network Spice, NX and VNC reaches 11%, 17% and 69% video quality respectively for the medium bitrate video. Lowering the bitrate gives Spice and VNC better quality whereas NX shows only a slight improvement. The numbers are 21% for Spice, 18% for NX and 82% for VNC.

### 6.2.2 Data usage

Because the baseline is the same for medium and low bandwidths, please refer to section 6.1.2 for information about it.

When viewing the medium bitrate video over a highly loaded network both Spice and NX are within 1% of the data used compared to the baseline at 1409 MB and 316 MB as seen in figure 7. VNC however uses 7% less data at 196 MB. Having no added load on the network makes Spice drop the data transmitted significantly to 185 MB. NX and VNC are within 1% of the baseline with 316 MB and 209 MB respectively.

With the low bitrate video, Spice and NX are within 1% of data transmitted compared to baseline for both high and low load on the network. 1284 MB and 1277 MB are transmitted for Spice during high and low load while NX sends 282 MB of data for both. VNC sends 173 MB and 179 MB of data for high and low loads respectively which is within 5% of baseline data transmitted.

## 7 Analysis

### 7.1 Spice

It is clear to see from the data shown in section 6 that Spice performs the worst of the three protocols. Not only does it have the worst video quality for all but one combination of variables, it only shows some major improvement when going from medium to low bitrate during no added network loads in a medium bandwidth network. This improvement is close to twice the quality of the same test but with a medium bandwidth video. Even so, the quality is still worse than VNC's quality in the worst conditions in a medium bandwidth network.

It is likely that the high amount of data transmitted is what gives Spice the low video quality. With 80 times the data of the source material, a good connection is required even when using a LAN network. At 1404 MB in 23 seconds, close to 500 Mbit/s is needed to be able to view the video in ordinary speed without losing any data.

### 7.2 NX

NX seems to be the most stable of the three protocols. It achieves data transmissions close to the baseline and shows no large difference between medium and low bitrate videos. The difference between high and no added loads however are quite big. 10 times higher quality in a low bandwidth network and around 2 times the quality in a low bandwidth network. Even so, the quality of the video is quite bad, it never reaches above 20%.

### 7.3 VNC

VNC has the best video quality overall with a peak of 82% at the best conditions. When lowering the bitrate of the video, the video quality consistently improves around 18%.

VNC has the lowest amount of data transferred which likely makes it easier to achieve its high video quality. Even so, it can be seen that VNC suffers in a low bandwidth environment where it only sent around 60% of the data compared to the baseline.

### 7.4 Related work

When comparing the results to other work such as by Baratto et al. (2005) it is clear to see that NX has made few improvements. In those test it reached 12% quality compared to the 17% under optimal conditions in this experiment. In the experiments done by Baratto et al. (2005) NX was the protocol that performed worst, this is no longer the case as Spice performs worse than NX in all configurations but one. VNC has made great improvements, climbing from around 23% to 82%. This could be attributed to the fact that TigerVNC is used instead of RealVNC as in the experiments by Baratto et al. (2005). TigerVNC has many improvements that focus on running 3D and video applications that RealVNC lack.

## 8 Conclusion

The results obtained in this work indicate that Spice does not provide the high quality user experience the creators of Spice claim in regards to viewing multimedia content over a remote desktop protocol. It is likely that the lack of quality comes from the high data usage. Instead, VNC is the remote desktop protocol with the best video quality. There are however some issues with this work as discussed below.

### 8.1 Discussion

Even though the results presented give an indication on how the remote desktop protocols that are examined perform, these results may not be an accurate description of their quality. If a protocol heavily compresses the data both in the slow-motion and ordinary speed trials it can achieve a high quality rating even though the video quality would be dissatisfying. Comparing the percentages of different protocols is problematic. Two protocols that achieve the same percentage could have very different quality. Because of this, the results from this work should only be seen as an indication of how the protocols can perform. A more suitable way to measure quality can be to do the experiment with a more subjective measurement of quality. For example, showing the videos to people and letting them use a scale to give a quality measurement. This could be combined with the slow-motion benchmarking to see if the quality is comparable.

While watching the videos being shown in the experimentation, it became apparent that the video quality did not change much. The thing that changed was how long it took to play the video. As available bandwidth lowered, either from limiting the speed of the link or by introducing load, the time it took to play the video increased. This made the videos unwatchable as some were played for several minutes while the source material was 23 seconds long. Some setups played the video for as long as 50 minutes. This leads me to believe that the protocols do not adapt when conditions change. A protocol could adapt to less available bandwidth with, for example, increasing the amount of compression used. NX and VNC have settings for quality that can be set manually whereas Spice detects available bandwidth when the connection to the server is made. However, none of the protocols automatically detect when these conditions change.

Playing a video is only a part of how a user experiences remote desktop protocols. Different applications call for different settings in the remote desktop protocols to achieve a good user experience. For example, working in a word processor requires less frequent updates than scrolling down on a web page. Because these different environments require different amounts of bandwidth, this is another reason for having automatic adjustment of quality settings in a remote desktop protocol.

#### 8.1.1 Ethical aspects

The availability of remote desktop protocols makes it possible to work from home. However, there are implications of this. When deployed in a teleworking environment, the person working from home gets a lot of responsibilities. Even though the person might not be distracted by the office environment, working from home has distractions too. It is up to the employee to make sure that as much time is spent actually working as if he would be working in the office. According to Fairweather (1999), one



of the recognized problems associated with teleworking is that the people working from home may feel isolated. They state that a method to counter this is to use video-conferencing. However, they go on to state that the employee might feel that this is an invasion of privacy for the whole household as people or items that the person wants to keep private may be seen by the employer.

### **8.1.2 Social aspects**

The purpose of this work is to find out if Spice has the high quality user experience that they claim. In light of the results this might not be the case. Because of this, administrators should take extra care when choosing the remote desktop protocol that suits them best. This could affect the community as this might take a lot of time and an alternative remote desktop protocol might cost money. In the case of teleworking, Fairweather (1999) identifies interception by third parties as a potential problem. If an insecure connection is used, sensitive information may be intercepted. Fairweather (1999) states that even though the teleworker's own work might not be sensitive, there is always a possibility that other employees and business associates may send them sensitive information unaware that the teleworker is using an insecure connection.

## **8.2 Future work**

There are several ways to improve on this work, three of them are identified below.

### **8.2.1 Web application**

One of the partial goals was to test the protocols in a web environment. However, due to lack of time this could not be completed. As using multimedia is only a part of the user experience of a remote desktop protocol, this test gives another indication of how the protocols behave in a real environment.

### **8.2.2 Voice over IP quality measurements**

As discussed in section 8.1, a different way to measure quality could improve on this work. Borrowing from voice over internet protocol (VoIP) quality measurement could be a way to do this. An example of this is using mean opinion score (MOS) where human judges rate samples on a scale from 1 to 5. By using this scale, a subjective score is achieved that can be used by itself or compared to another method such as slow-motion benchmarking quality.

### **8.2.3 Different configurations**

The protocols examined in this work all have settings that affect their use. Due to lack of time, most of them had to be ignored. In future work, these could be examined to see if they provide a better user experience. Some examples of this include quality settings for the remote desktop protocols and monitor resolution.

There are also several variables that relate to the experiment environment. For example, working with different operating systems or different network bandwidths and computer hardware. The experimentation in this work was done in a Linux environment. Other operating systems such as Windows should also be taken into account.

## References

- Baratto, R. A., Kim, L. N., & Nieh, J. (2005, October). THINC: a virtual display architecture for thin-client computing. *SIGOPS Oper. Syst. Rev.*, 39, 277–290.
- Deboosere, L., De Wachter, J., Simoens, P., De Turck, F., Dhoedt, B., & Demeester, P. (2007, june). Thin client computing solutions in low- and high-motion scenarios. In *Networking and services, 2007. icns. third international conference on* (p. 38).
- Fairweather, N. B. (1999). Surveillance in employment: The case of teleworking. *Journal of Business Ethics*, 22, 39-49. Available from <http://dx.doi.org/10.1023/A:1006104017646> (10.1023/A:1006104017646)
- Harpaz, I. (2002). Advantages and disadvantages of telecommuting for the individual, organization and society. *Work Study*, 51(2), 74-80.
- Lubonski, M., Gay, V., & Simmonds, A. (2005, oct.). A conceptual architecture for adaptation in remote desktop systems driven by the user perception of multimedia. In *Communications, 2005 asia-pacific conference on* (p. 891 -895).
- Miller, K., & Pegah, M. (2007). Virtualization: virtually at the desktop. In *Proceedings of the 35th annual acm siguccs fall conference* (pp. 255–260). New York, NY, USA: ACM.
- Nieh, J., Yang, S. J., & Novik, N. (2003, February). Measuring thin-client performance using slow-motion benchmarking. *ACM Trans. Comput. Syst.*, 21(1), 87–115.
- NoMachine. (2010). *NoMachine Makes First NX 4 Technology Preview Available*. [Accessed 2012-05-16]. Available from <http://www.nomachine.com/news-read.php?idnews=331>
- NoMachine. (2012). *Introduction to NX technology*. [Accessed 2012-05-16]. Available from <http://www.nomachine.com/documents/intr-technology.php>
- Red Hat, Inc. (2009). *Spice for newbies*. [Accessed 2012-05-16]. Available from [http://spice-space.org/docs/spice\\_for\\_newbies.pdf](http://spice-space.org/docs/spice_for_newbies.pdf)
- Richardson, T., Stafford-Fraser, Q., Wood, K., & Hopper, A. (1998, jan/feb). Virtual network computing. *Internet Computing, IEEE*, 2(1), 33 -38.
- Sikes, N., Mason, K., & VonLehmden, S. (2011). Telecommuting advantages and challenges for IT management and staff. In *Proceedings of the 49th sigmis annual conference on computer personnel research* (pp. 20–25). New York, NY, USA: ACM.