IEG3830 Product Design Project

BENCHMARKING ON REMOTE DESKTOP APPLICATIONS

BY

Chau Man Ping Grace (02501421)
Hui Ka Hung (02533281)
Wong Tze Ming (02677401)
Ng Matsuba (02537733)

AN FINAL REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE IEG3830 PRODUCT DESIGN PROJECT
DEPARTMENT OF INFORMATION ENGINEERING
THE CHINESE UNIVERSITY OF HONG KONG

April, 2004

Abstract

Our focus in this final report is to determine the suitability, strength and weaknesses of remote desktop applications operating in a 2.5G/3G wireless environment. The applications evaluated include Remote Desktop and Remote Desktop Window of Window XP Professional, VNC, TightVNC and Citrix. The result shows that the remote desktop applications continue to require a high bandwidth environment in order to operate effectively. The factors affecting the performance will be discussed, which include: optimization of various compressions, caching, and event handling techniques etc.

Table of Contents

Session 1

Introduction

1. Review of our last report

In our last report, we have already discussed why we have this benchmarking, the special features of our benchmarking and the experimental set up etc. Let's do some review here.

1.1. Why we have this benchmarking

1. Remote desktop applications are becoming more and more popular recently
2. Transmitting data in wireless environment is an emerging trend.
3. Therefore having remote desktop applications run on wireless devices may have great market value.
4. Thus we want to find out the possibility of this idea – running remote desktop applications on wireless devices. We do benchmarking on different remote desktop applications in a wired environment. We try to mimic the wireless environment by lowering the bandwidth using special setup.

1.2. Special features of our benchmarking.

1. The testing bandwidth will range from 100Mbps to 8kbps
2. Our tests cover many areas. One of them is on multimedia – Flash and Java Applet.
3. The other area that our tests focus on is messaging applications – E-mail and ICQ.
4. Besides testing the RDA in a full screen environment, we will also perform the tests in a smaller resolution (130 * 150).
5. Before starting the experiment, we perform a lot of trial tests to find out the optimal conditions for each RDA.
6. As our main concern is on the transmission efficiency, we also have the analysis on the percentage of the data packets versus the total packets
7. China will be one of the biggest markets. Thus our tests include the display of Simplified Chinese website.

1.3. Experimental Setup

1. Tests designed:

There are 4 kinds of screen updates in our tests:

I. Scrolling of Text
II. Scrolling of Picture
III. Static Picture
IV. Animation

Also there are 2 kinds of user inputs in our tests

I. Keyboard sequences
II. Mouse sequences

A table showing the types of data we collected in different tests:

		Type of information				Types of feedback	
		I.	II.	III.	IV.	I.	II.
1.	E-mail (attach pic)-	☆	☆			☆	☆
2.	E-mail (text)	☆				☆	
3.	Excel	☆		☆		☆	☆
4.	Flash (30s)				☆		☆
5.	ICQ	☆				☆	
6.	Java (stock chart)			☆			☆
7.	Pictures		☆			☆	
8.	PowerPoint	☆	☆	☆	☆	☆	☆

9.	SCMP	☆	☆			☆	
10.	Word	☆	☆	☆		☆	
11.	Xinhua	☆	☆			☆	

2. Benchmarking tools:
   I. EZ Macro – unifies the keyboard and mouse input throughout all tests
   II. Ethereal – acts as a network monitor
   III. Net limiter – controls the bandwidth of the channel.

For the detail experimental setup as well as the optimal settings of each application, the reader can refer to our initial report.

1.4. Types of Tests

The tests lie in two main categories, one is to compare the performance on high bandwidth (observe the efficiency of their algorithm), and the other is to observe the techniques used by the RDAs in dealing with a low bandwidth environment, possibly with interference.

1.4.1. Comparing efficiency of algorithm

The test proposed are mainly based on the 4 kinds of screen update mentioned, and based on the popularity. Followings are the description of the tests:

I. Microsoft Office
It includes Microsoft Word, Excel Power Point.
II. Web Page Test
It includes Pictures, XinHua net, South China Morning Post online, SINA simplified (China), Flash Movie, and Java Applet.
III. Other online applications
It includes Email (with and without picture) and ICQ.

1.4.2. Working in Low Bandwidth

In the real world, the wireless communications environment may not be stable. We try to mimic this environment by performing tests on low bandwidth environment. This is the worst working conditions, and the number of packet lost might reach an unacceptable level.

So, we proposed 5 tests on the worse case (8kbps). They are

(labels is same as1.3's table)		Time (sec)	Type of information				Types of feedback	
			I.	II.	III.	IV.	I.	II.
1.	E-mail (text)	60.0	☆				☆	
2.	Flash (30s)	30.0				☆		☆
3.	ICQ	18.0	☆				☆	
4.	Pictures	20.0		☆			☆	
5.	Xinhua	20.0	☆	☆			☆	

The types of feedback have shifted to focus on keyboard sequence. In a much slower tempo than the test we performed on 100Mbps & 56kbps. Meanwhile, the types of information have shifted to focus on test and static pictures, as we will not expect dynamic pictures are still affordable from user's point of view.

Session 2

Analysis

2. Analysis of the data

In this report we mainly focus on the data analysis part. We will investigate how the applications react differently upon the changes in bandwidth. Factors like average bytes transferred, time for completion, no. of packets sent, leakage, caching effect will be discussed.

2.1 Remote Desktop Connection of Window XP Professional

2.1.1. Introduction

RDP (Remote Desktop Protocol) 5.1 [1] is employed in the Remote Desktop Connection of Window XP Professional. RDP is designed to provide remote display and input capabilities over network connection for Window based applications running on a server. It was first introduced in the Window NT Server 4.0, with RDP version 4.0. RDP is actually based on an existing ITU T.120 family of protocol [2] – a data protocols for multimedia conferencing. The interoperability of basic functions which are required by remote desktop applications is ensured by the Recommendation of T.120 protocol. These functions include file transfer, still image exchange and shared whiteboards. The T.120 protocols also allow many forms of Data/Telemetric information in telecommunicating between several multimedia terminals and it is suitable for use on many types of network.

There was a lot of improvement from RDP 4.0 to RDP 5.1. Additional features such as Virtual Channels $_{(a)}$, Compression Enhancements, and Persistent Bitmap Caching $_{(b)}$ are introduced. These features are indeed helpful in enhancing the performance over other applications such as RealVNC.

Remark (a): Using the RDP Virtual Channel Architecture, existing applications can be augmented and new applications can be developed to add just about any feature that requires communications between the client device and an application running in a Terminal services session.

Remark (b): Persistent bitmap caching augments the RAM cache with a 10MB disk cache for bitmaps. Bitmaps that get cached in memory can also be stored in the persistent bitmap cache.

2.1.2. Tests Result

I. in LAN environment – 10-100Mbps

Firstly, let’s look at how Remote Desktop Connection performs in LAN (100Mbps) environment. Most applications run smoothly without any noticeable delay. The average bytes transferred increases with the amount of screen updates. (Fig. 1) In displaying the content, the video driver on the server side will construct the rendering output information into the net packets. [1] The client will receive the rendering data and interpret them into the corresponding Win32 GDI API calls. [3].

In Word and Excel Tests, which involve typing and scrolling text, the bytes transferred are only 62kBytes and 90kBytes. For tests with more pictures and more screen updates, like the Web Page tests (Pictures, Xinhua etc.), the average bytes transferred increased to 1000-2000kBytes. In the most extreme case Flash, which has screen update almost every millisecond, the average bytes transferred is 10,000kbps. In this case, a little latency can be observed and the movie is no longer continuous. This also proves that the latency is directly proportional to the average bytes transferred.

The no. of packets transferred is also directly proportional to the average bytes transferred (Fig.2). This implies the bytes per packets in different tests are more or less the same, with several hundreds bytes per packet. The no. of data packets are 60-70% to the no. of total packets. This remains constant through out all tests.

The effect of caching becomes apparent when the tests are repeated. (Fig.3) Since RDP4.0, it has already implemented a glyph and fragment caching mechanism at the client. [4] The glyph and fragment are cached in the client's RAM. When the strings needed to be displayed again, the server does not need to resent the same glyphs, instead it will tell the client to reuse a cached glyph or fragment. In addition to that, RDP 5.0 adds persistent bitmap caching, which augments the RAM cache with a 10MB disk cache for bitmaps. Bitmaps that get cached in memory can also be stored in the persistent bitmap cache. Thus the performance is getting better and better when the tests are repeated – with less and less latency. However in the Flash test, the cache buffer may not be able to accommodate such huge amount of data. Thus in the Flash test, the caching is not effective. However due to experimental errors for very few tests, the bytes transferred increased a little bit when the tests are repeated in the 2nd or 3rd time. (Fig.3). These experimental errors include noises, periods between tests vary etc.

II. In modem environment – 56kbps

In 56kbps, the performance in different tests diverges. (Fig.1) For Word and Excel test, the performance is still satisfactory without any noticeable delay. However for Web Pages test, the latency becomes obvious – 70-80 seconds are needed to finish a 60 seconds test. (Fig. 4) This is because in Word and Excel test, the data are transferred as glyph and fragment; but for Web Page tests, the data are transferred as blocks of bitmaps. The sizes of glyph are much smaller than bitmaps and the glyph can be reused. Thus in Web Page tests, the server is sending data much faster than the channel can cope with. Queuing delay arises and therefore latency is observed. This is the most serious in the Flash test. 960 seconds is needed in order to finish a 60 seconds Flash test, which is rather unacceptable.

In Web pages tests, it is observed that the content is displayed grid by grid, in a horizontal manner. During scrolling, although the content in the whole screen is changed (move up or down) only the content that is scrolled in will appear grid by grid. This proves that the RDP client is smart enough to scroll the content upward/download and the server will only send the scroll in part, instead of sending all data in the screen once again to the client.

Interestingly enough, the average bytes transferred this time is much smaller to the 100Mbps case, in spite of the same tests are done (Fig 1). This is due to the bandwidth reduction features of the RDP. [5] It supports various mechanisms to reduce the amount of data transmitted over different network connection. This mechanism include the caching of glyph and fragments in RAM and persistent bitmap caching. In additional to this, the algorithms for determining when to send screen updates to the client were optimized to increase the responsiveness of applications over low bandwidth connections. Also the RDP client was tuned for reducing CPU utilization when rendering output. A combine effect is the reduction in network traffic as well as the reduction in CPU utilization.

On the other hand, the caching effect brings more enhancements in such low bandwidth environment. The average bytes transferred decreased when the tests were repeated (Fig. 3). Latency almost disappears if a web page is browsed over 10 times. However the bytes transferred decreases less significantly than in the 100Mbps case. This is due to the fact that the compression and caching has already reached its limit.

III. Worst case – 8kbps

In order to bring out the worst side of the applications, we have performed the tests in 8kbps environment. (Fig 4) Amazingly enough, the E-mail (text) tests are still running smoothly and credits must be given to the glyph and fragments caching. The typing latency is surprisingly small, which made word processing still possible in such low bandwidth environment. Compare to RDP 4.0, RDP 5.1 had all of the components in the round trip typing path being tuned, this effectively reduced the round trip input latency by 50% with an 80% reduction in bandwidth. [1]

However in the Web Pages and ICQ tests, the performance is poor. Since the bandwidth is so low, the data rate of the server will surely be much higher than the channel bandwidth. Queuing delay is significant thus the latency is extremely huge (Fig.5) – when the web pages are scrolled down, the content scroll in pixel by pixel, instead of inch by inch in the normal case. This is because RDP opts to output all the update content, regardless of the latency. On the other hand, RealVNC or TightVNC will skip screen updates in order to synchronize with server. Therefore compare to VNC, RDP is less desirable in this case as the users usually prefer to skip screen updates.

For the Flash test, Remote Desktop Connection disconnects after about 3mins. The data rate of the server will exceed the bandwidth by a very large extent. Queuing delay may be too huge thus packet loss may occur. Too much packet loss may result in disconnection in this case.

IV. Charts

Fig 1 Average Bytes transferred in different tests (100Mbps, 56kbps)

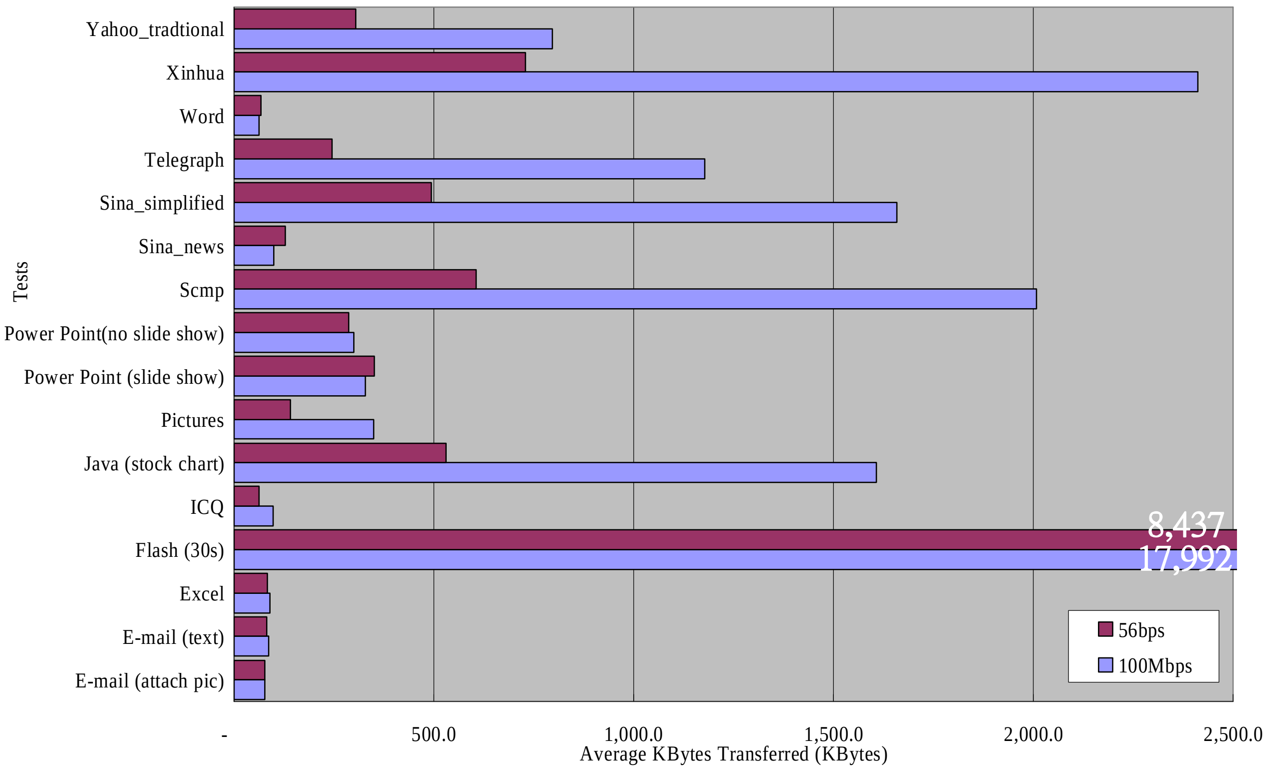


Fig. 2 Number of packets transferred in different tests (100Mbps, 56kbps)

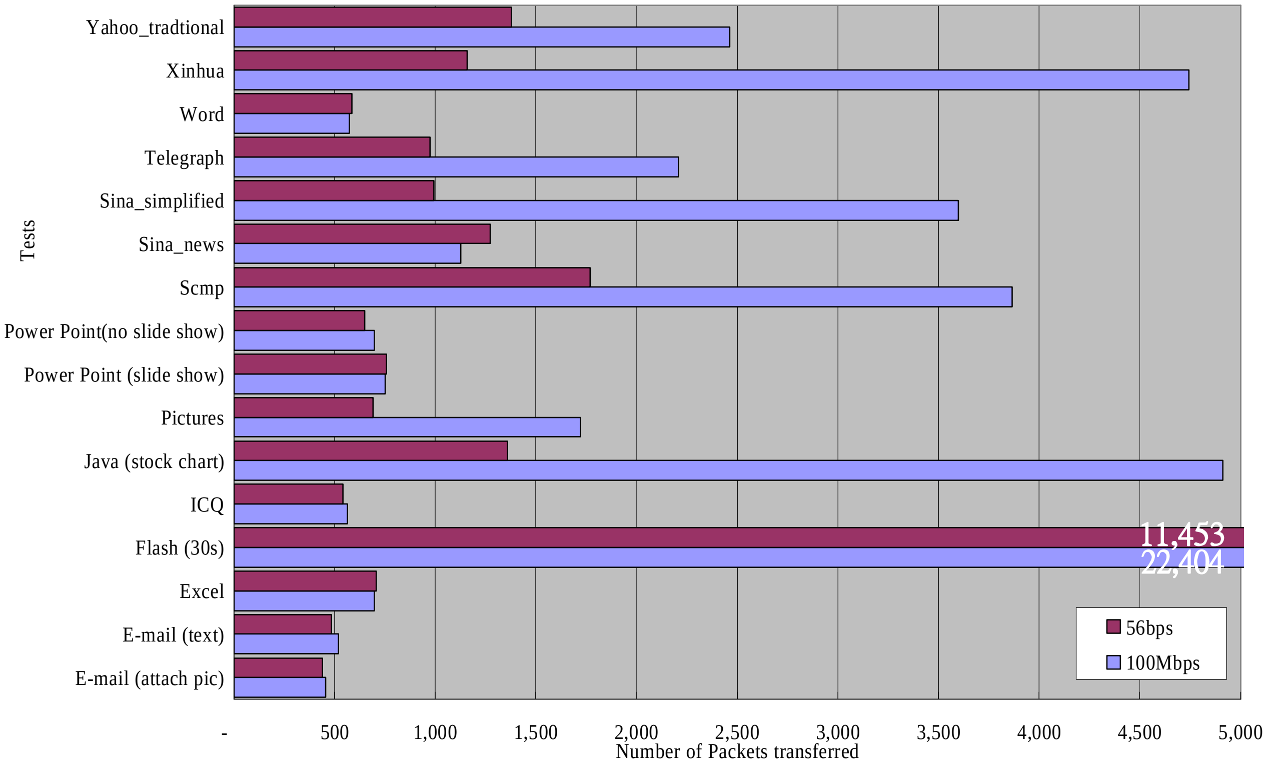


Fig. 3 KBytes transferred for the same tests during different times (100Mbps, 56kbps)

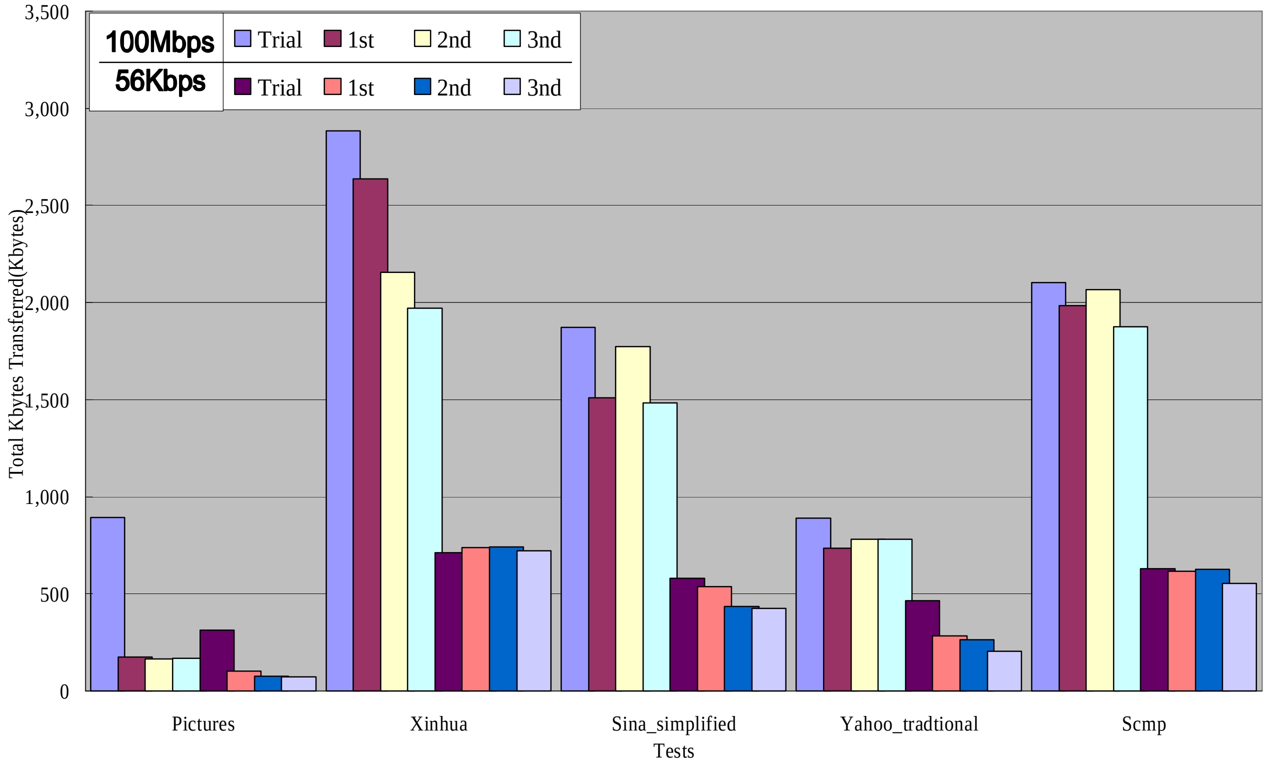


Fig. 4 Time for completion in different tests (100Mbps, 56kbps)

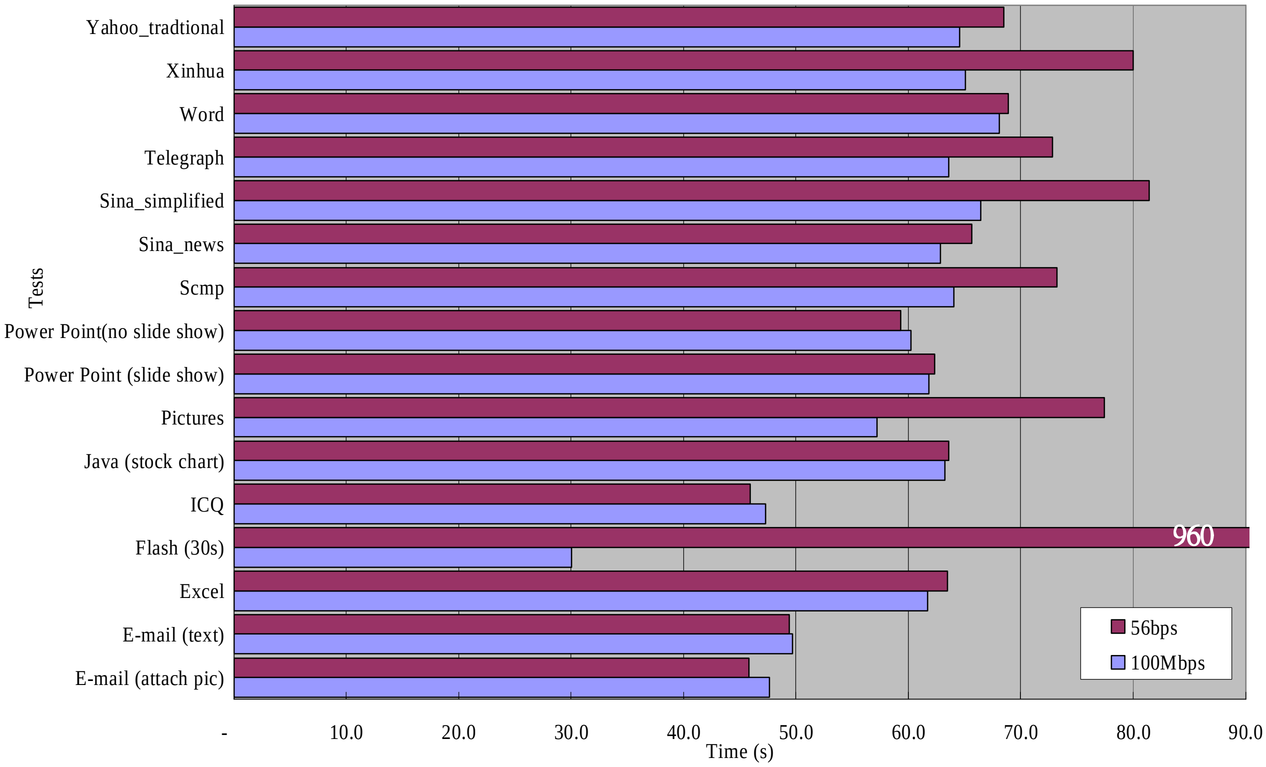
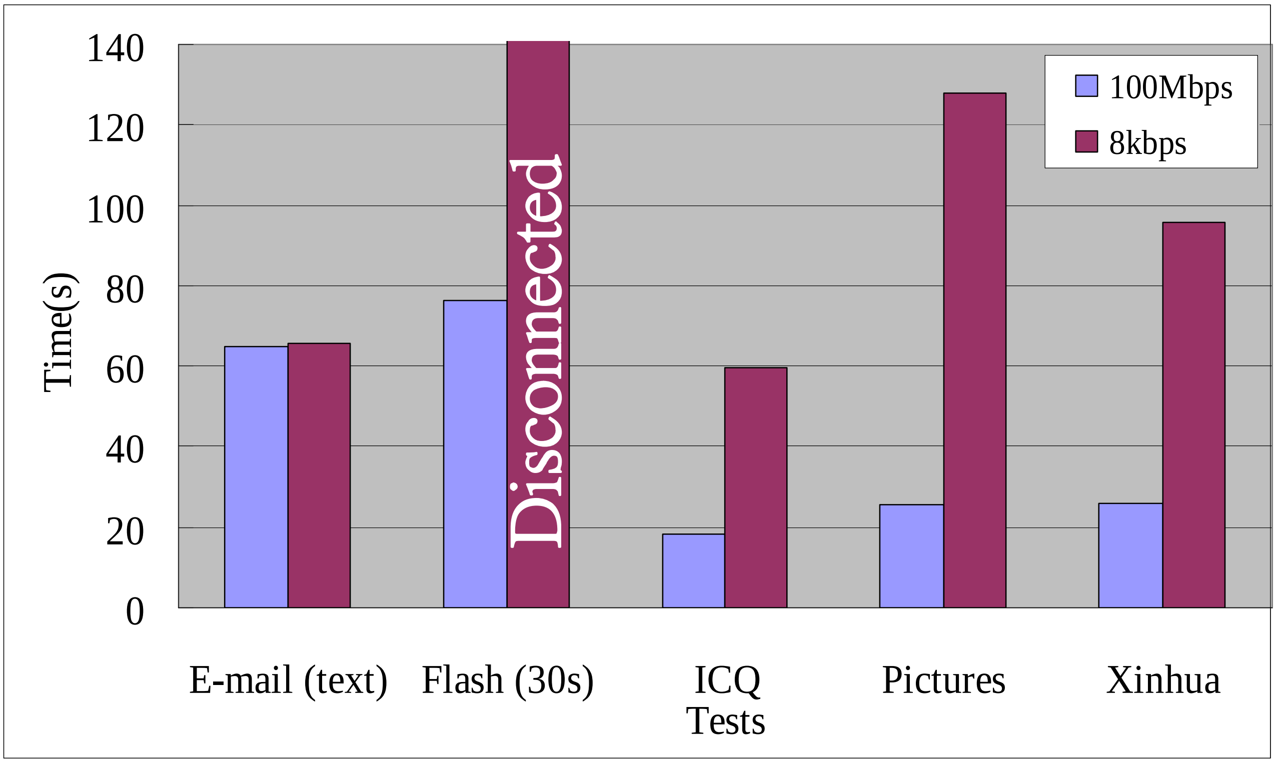


Fig. 5 Time for completion in different tests (10Mbps, 8kbps)



Remarks: The tests taken here is specially designed, and are different from previous test which have the same name

2.2 Remote Desktop Web Connection

2.1.1. Introduction

Remote Desktop Web Connection is a developed version of Terminal Service of Windows NT. It uses an ActiveX control, which is downloaded automatically, to connect to the Server where a RDP service is running. Therefore, it shares similar properties with RDP.
The similarities are as follow:

1. No Frame Skip
   The frame update does not have any Frame Skip, it has to transmit the whole frame from the server to client before accepting the next event.
2. Virtual Driver for Display
   Instead of using a screen capture mechanism, RDP or RDP web use a virtual driver to capture the screen, which means that the driver of the server will construct the rendered information and reconstruct the screen on the client by a Win32 API call.
3. Automatic Bandwidth detection
   It behaves differently on different bandwidth, the bandwidth status is detected and different algorithms are used for different condition.
4.

However, there is some differences between RDP and RDP Web:

1. Version Different
   RDP web use RDP 5.0 while RDP use RDP 5.1 for the protocol. Since the protocol is backward compatible. It does not cause much different to the user. The different between RDP 5.0 and 5.1 is just some optimizations.
2. No Sound Support
   Unlike the RDP, RDP web does not support the sound support, which means that the data transfer is a little different for a data with sound and a data without sound.
3. No local drive mapping
   RDP web does not support the local drive mapping which is used for file transmit on the RDP connection. It is not support by the RDP web.

2.2.2. Tests Result

I. in LAN environment – 10-100Mbps

The standard bandwidth is 10 to 100 Mbps with average Round Trip Time ("RTT") less than 10ms. All applications can run smoothly on this channel. However, the data transfer size varies from several Kilo Bytes to tens Mega Bytes for testes run for 60 seconds, depending on the frequency on screen update, the nature of data shown and the updated screen size. The data transfer depends on the frequency of screen update, for example, a Flash Movie need to transfer for about 10MB while a Word document editing transfer only tens of KB (fig. 6). It depends also on the nature of the data shown, a web data with pictures needs to transfer more data than that of a Word application, for a Web with more text like Xin Hua, it needs to transfer more data than that of SCMP which have less text. The data transferred also depends on the screen size update, the larger the screen size update, the more data sent, even for a whole screen display, it just update the parts changed.

It does not show significant delay in the LAN environment, each key press or mouse action can response fast enough with screen update. It is as expected as the RTT is not too long.

RDP Web has powerful caching techniques, when we try to carry the same experiment for 4 times, the data transferred drops can be as large as 90% for Word and Excel and about 30% on average for each application. However, the performance of caching does not perform on Flash movie. Also, the RDP is smart enough to detect the scrolling bar event, which it saves a large amount of bandwidth.

The time of completion (fig 8) is nearly the same as that of the test performs. By the above factors, RDP Web can run even for multimedia purposes in a LAN environment.

II. In modem environment – 56kbps

In a Modem environment, the bandwidth is limited to a bandwidth similar to a GPRS system (64Kbps), the delay is mainly due to the RTT which is not so significant. In this environment, most static applications can run smoothly, while some dynamic applications like Flash Movie run with delay. Amazingly the static applications like Word, Excel and Email run as smooth as that of LAN environments while web pages have acceptable performance.

The time of completion for the whole process has increased less than 10% than that of the finishing time for the static applications, after several caching, nearly no increase in the testes. However, for those dynamic applications, the time of completion increase sharply, the movie that play for 1 minutes but the completion time will end with more than 10 minutes. (fig 8)

The effect of caching is significant, it helps a lot for the static application, even in such a low bandwidth, we can do the application smoothly. The data transfer drops sharply, as much as 90%. However, for the dynamic applications, the effect is not so significant. There is just 10% drop in the data transfer.

The delay is acceptable that when moving the mouse, the responses of the cursor is done within 1 second (without accurate measurement) which is acceptable to most applications that office use.

As a conclude, for the RDP Web using over GPRS is still acceptable for that of office applications and suitable for office use.

III. Worst case – 8kbps

A standard GSM or PCS mobile phone system has a bandwidth of 9.6Kbps. When we limited the bandwidth to such a low bandwidth, all the applications become very low performance when some may even fail to test. Connection establishing screen only load up by chance, loading up of the first desktop screen use more than 2 minutes. The connection may even drop due to the insufficient bandwidth, and the delay is significant in the fact that the mouse follows the movement in the client side after 3 to 4 second.

The performance of the RDP web in this environment is totally unacceptable to users. It cannot run in such a low bandwidth.

IV. Charts

Fig. 6 Average Bytes transferred in different tests (100Mbps, 56kbps)

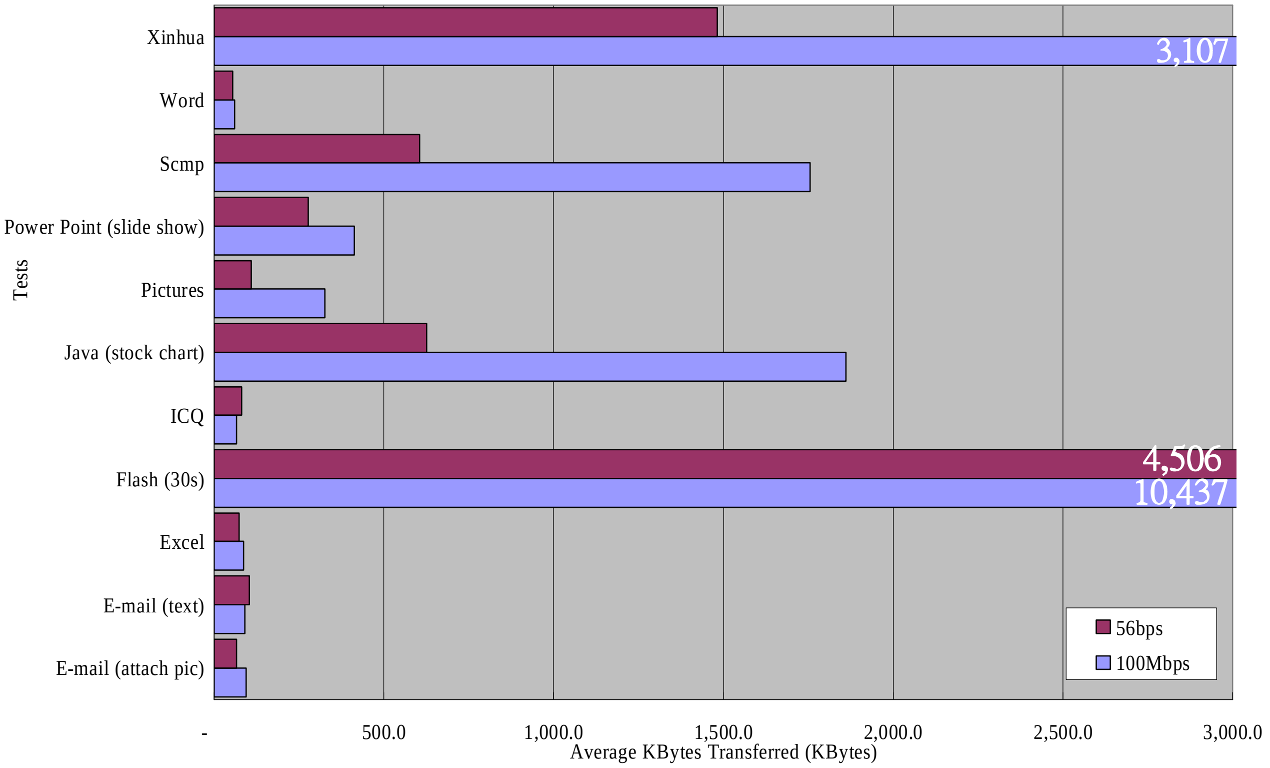


Fig. 7 Number of packets transferred in different tests (100Mbps, 56kbps)

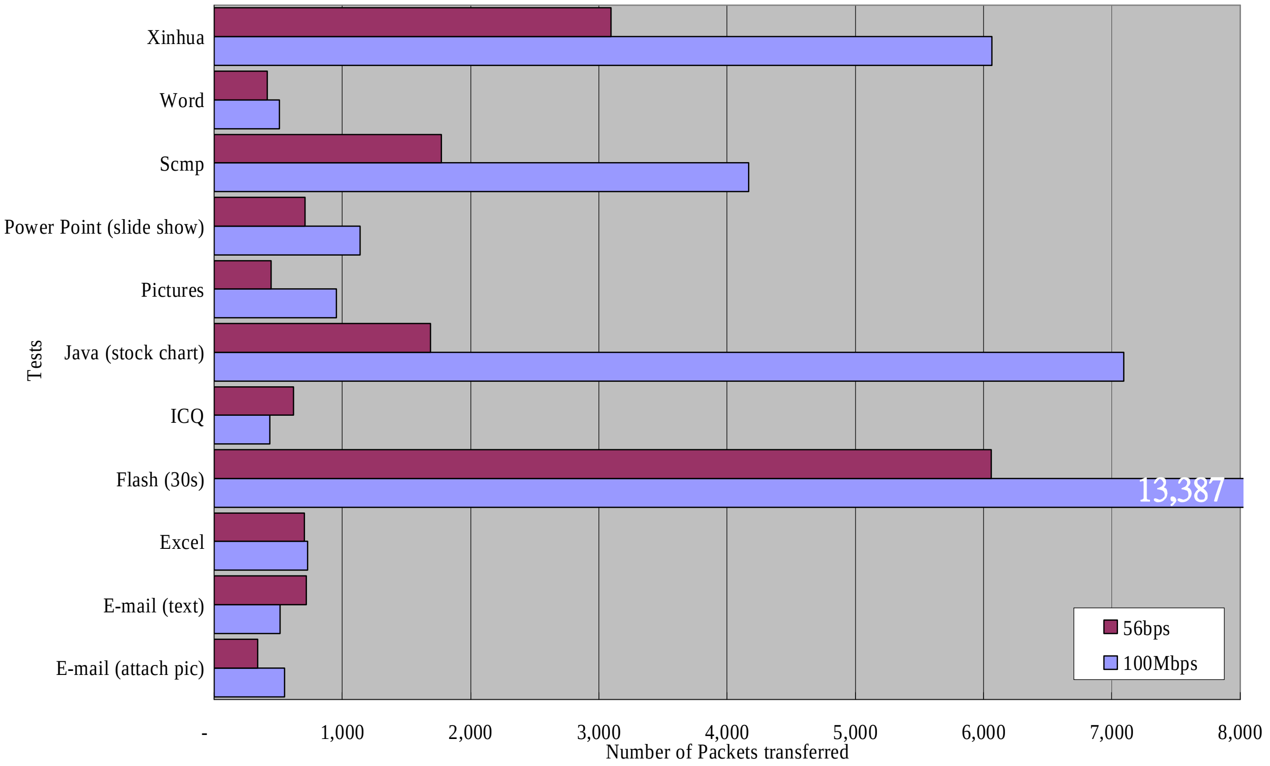


Fig. 8 KBytes transferred for the same tests during different times (100Mbps, 56kbps)

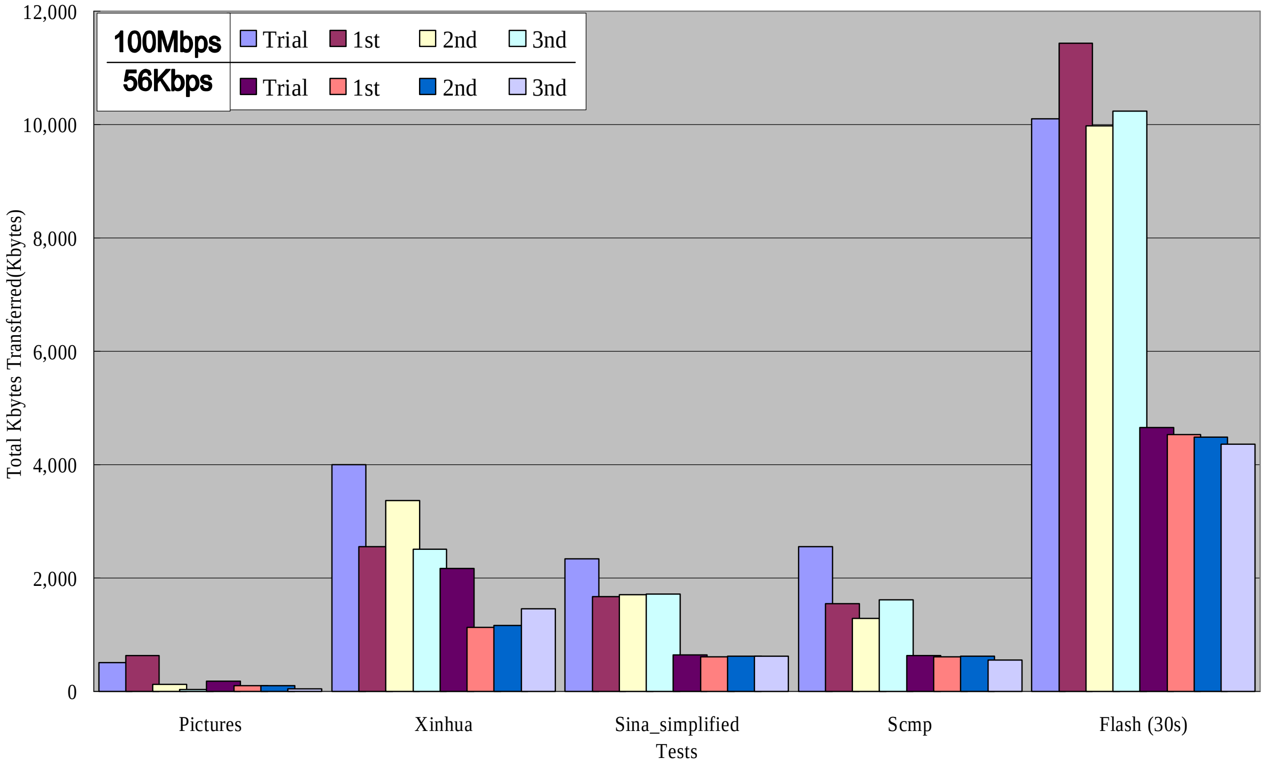


Fig. 9 Time for completion in different tests (100Mbps, 56kbps)

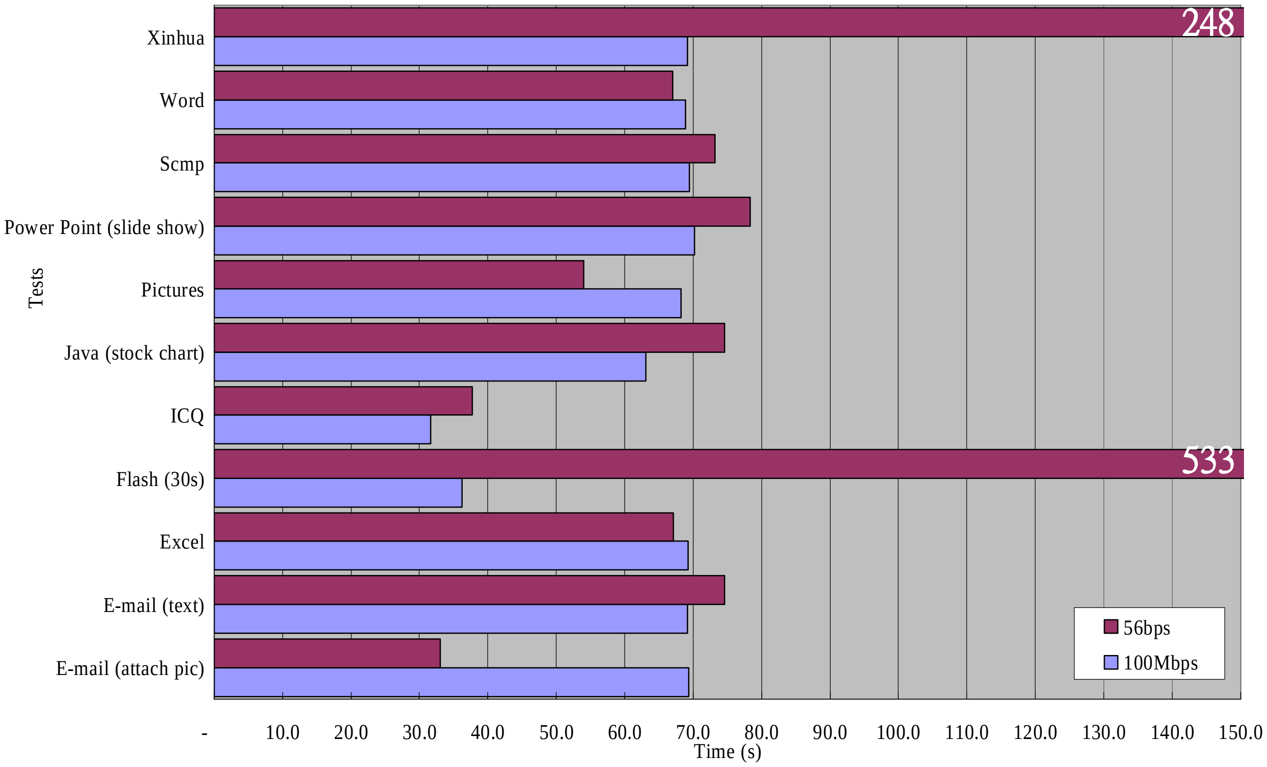


Fig. 10 Time for completion in different tests (10Mbps, 8kbps)

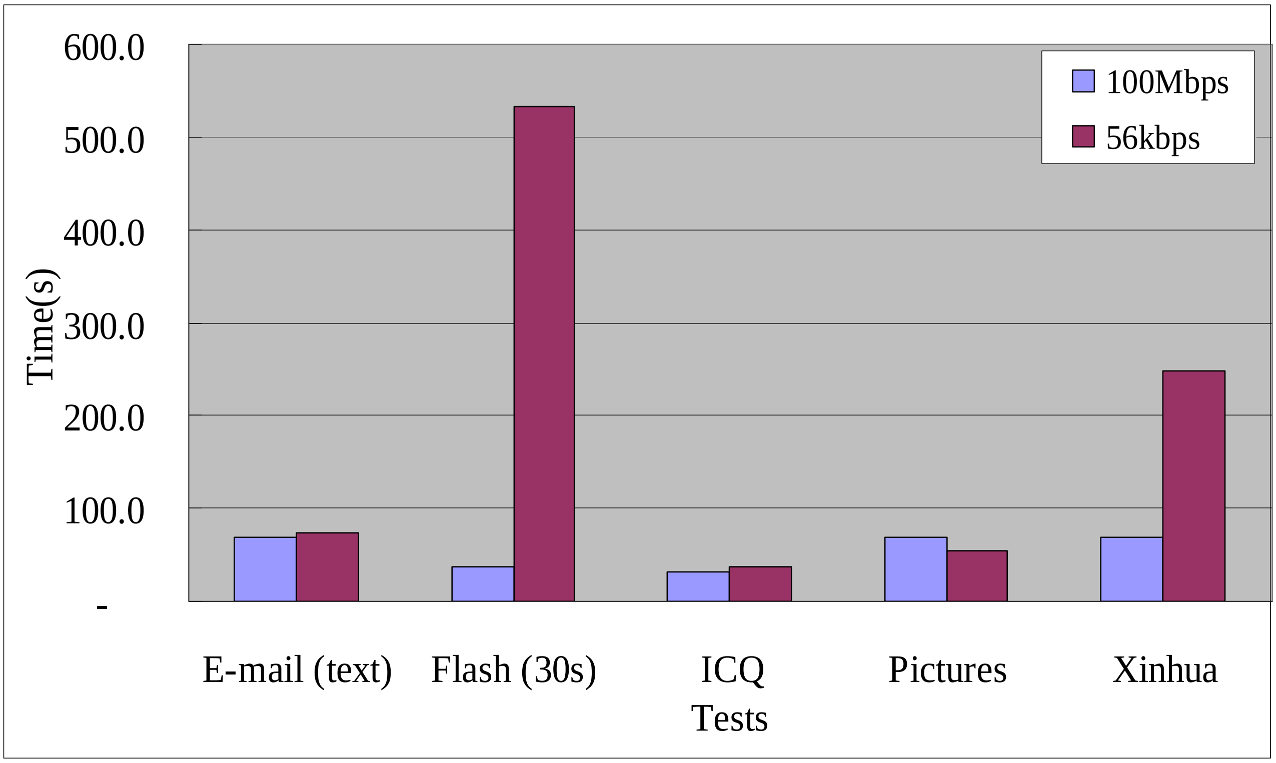
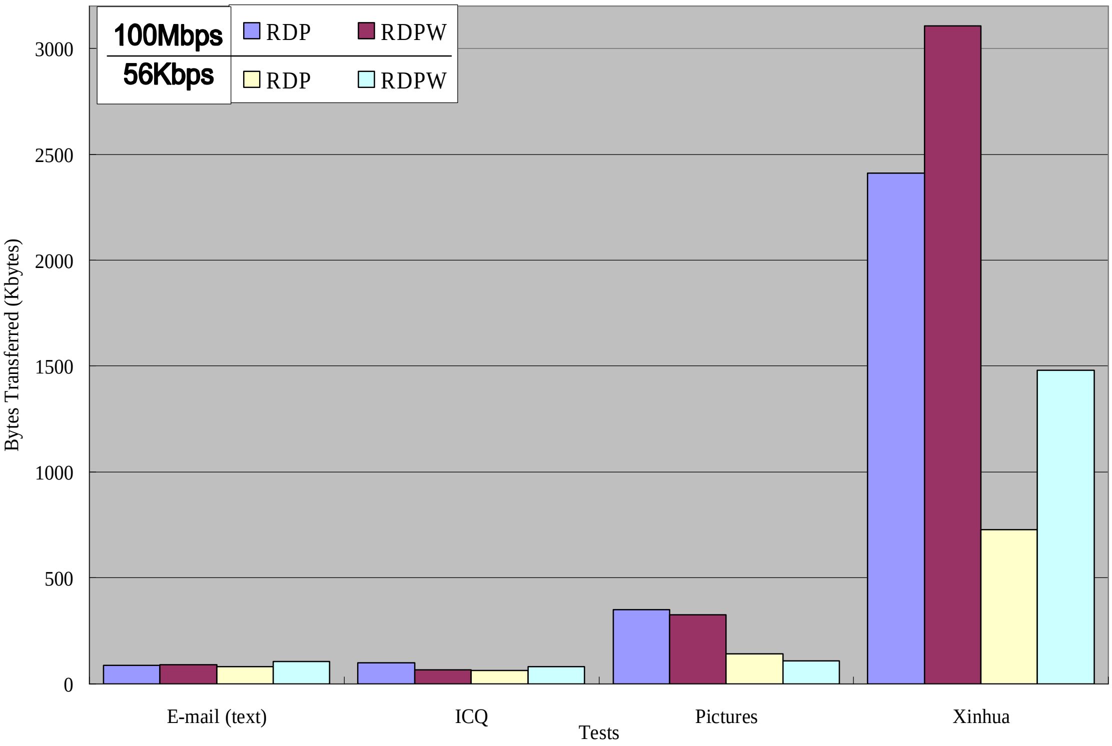


Fig. 11 Comparison with RDP in byte transferred of different test and different environment



2.3 RealVNC 3.37

2.3.1. Introduction

RealVNC (VNC is an abbreviation for Virtual Network Computing) is a client/server software package allowing remote network access to graphical desktops. With RealVNC, people can access their machines from everywhere provided that the machines are connected to the Internet.

RealVNC differs from other remote display systems in three crucial ways: cross-platform, small and simple, and free. First of all, with RealVNC, a desktop running on a Linux machine may be displayed on a Windows PC, on a Solaris machine, or on any number of other architectures. There is a Java viewer so that any desktop can be viewed with any Java-capable browser. There is a Windows server, allowing people to view the desktop of a remote Windows machine on any of these platforms using exactly the same viewer. The simplicity of the protocol makes it easy to port to new platforms. Besides, RealVNC is small and simple, e.g., the Windows viewer, is about 150 kilobytes in size and can be run directly from a floppy. The entire Java viewer is substantially less than 100 kilobytes and takes less time to download than the images on some web pages. Moreover, RealVNC is free and can be downloaded, used, and redistributed under the terms of the GNU General Public License. [6]

2.3.2. Tests Result

I. in LAN environment – 10-100Mbps

In LAN (100Mbps) environment, some applications run smoothly, with few of them start to skip some screen updates in spite of the high bandwidth availability.

In the cases of Word, Excel, Email and ICQ, which consist of little color variations, the updates as observed are quite smooth. The possible reason is that CopyRect encoding is enabled. CopyRect (copy rectangle) encoding “can be used when the client already has the same pixel data elsewhere in its framebuffer. The server simply sends an X, Y coordinate giving the position from which the client can copy the rectangle of pixel data. This means that operations such as dragging or scrolling a window, which involve substantial changes to the screen, may require fewer bytes.” [7] In these test cases, the background simply consists of a single color. By using CopyRect encoding, the amount of data sent can be considerably reduced. Therefore, the latency introduced would be small, and the updates appear quite smooth. However, the amount of data transferred is larger compared with RDP, RDPW and Citrix. (Table 4) This shows that RealVNC does not perform according to the claim.

In other cases like Powerpoint, Webpage, Picture, Java and Flash, the performance is worse than those mentioned above, in the sense that some snapshots of screen updates are skipped. This means when RealVNC detects that the bandwidth is not enough for transmission of all screen updates, it will skip over some of the screen updates and less screen updates will be sent. For example, in Powerpoint and Java, what can be seen is only the result of animations rather than the whole animations. As another example, in Webpages, the scrolling of a window as observed is not smooth. By doing so, the amount of data transferred can be reduced and the latency observed will be smaller.

In spite of skipping some screen updates, the amount of data sent for the cases of Powerpoint, Webpage and Picture is still larger than that for the cases of Word, Excel, Email and ICQ. (Fig. 11)

When the tests are repeated for four times, the results are rather similar. (Fig. 14) This shows that caching is not effective despite CopyRect encoding can be considered as some sort of caching. [7]

The delay observed is obvious, about 5-20 seconds. The delay introduced is so large that in general it is not acceptable. (Fig. 13)

II. In modem environment – 56kbps

In 56kbps environment, skipping of screen updates almost appears in all tests.

For the cases of Powerpoint, Webpages, Picture, Java and Flash, the skipping of screen updates is much more obvious. As the bandwidth available is insufficient compared with LAN environment, one can imagine that the screen is updated very slowly. In fact, the screen is updated block by block, as if it were composed of blocks of images. As a result, superposition of current screen and previous screen appears. This is not desirable to users.

Unlike RDP, when the content is changed by scrolling up and down, the whole screen is updated block by block. This suggests that RealVNC sends the whole screen again, without considering whether the previous screen consists of part of the new screen or not.

For Word and Email, RealVNC transfers the whole screen when scrolling occurs. So, the behavior of RealVNC in Word and Email is similar to that of Webpages, but with less severe skipping of screen updates.

The exception cases are Excel and ICQ. The reason for the behavior of RealVNC in Excel and ICQ in 56kbps environment is that in these cases, only a small part of the screen is updated. So the screen updates as seen are quite smooth.

Unlike Citrix MetaFrame, on client-side, RealVNC attempts to display the whole previous screen before starting to display the new screen. This introduces a large amount of delay. When viewing the screen on the client and the server simultaneously, we observe that the screen updates of the client lag behind that of the server.

The performance varies in different cases in terms of the time of completion. In cases like Email and ICQ, which involve small amount of data transfer, the latency observed is still small, at most 2 seconds more than the same test in LAN environment. For other cases like Webpages, the latency introduced increases by a larger extent, for example, about 80 seconds is needed to complete the macro that would finish in 65 seconds in LAN environment. (Fig. 13)

In 56kbps environment, we observe that the amount of data sent is considerably smaller compared with LAN environment. (Fig.11) The amount of data sent in 56kbps environment varies from about 20% to about 85% of the amount of data sent in LAN environment. For small percentage drop in tests like ICQ, the amount of data transferred is small originally, so the screen can be updated with a small drop in the amount of data sent. For large percentage drop in tests like Webpage (in particular SCMP), the screen as seen on client-side only updates several times, this accounts for a large percentage drop.

III. Worst case – 8kbps

In this extremely low bandwidth environment, we expect that the performance of RealVNC would be even worse than that in 56kbps environment.

In the test cases like Flash and Webpage, the amount of data sent drops to about 2% of the amount of data that would be sent in LAN environment. (Table 4) The possible reason is that in LAN environment, little screen updates are dropped, but in 8kbps environment, the bandwidth available is so low such that much of the screen updates would be discarded, resulting in a large drop in the amount of data sent.

In the test case of Picture, the amount of data sent drops to about 55% of the amount of data that would be sent in LAN environment. (Table 4) The percentage drop is rather low, but this results in a relatively large completion time, about 140 seconds for the original 20 seconds macro.

When the time of completion is taken into consideration, we observe that the time needed increased dramatically, except for Email. (Fig. 15) The time of completion measured is at least three times the original time of completion of the macro; and in Picture, the time of completion increases to seven times the original time of completion of the marco. (Table 10) The performance in 8kbps is not acceptable in general.

IV. Problem of leakage

There is a problem in RealVNC. Even when the server/client is idle after completion of macro, there are still data transferring from the server to the client. We label this phenomenon as “leakage”. The amount of data transmission caused by leakage is rather constant for each test case and link bandwidth, about 20 kilobytes for each test.

IV. Charts

Fig. 11 Average Bytes transferred in different tests (100Mbps, 56kbps)

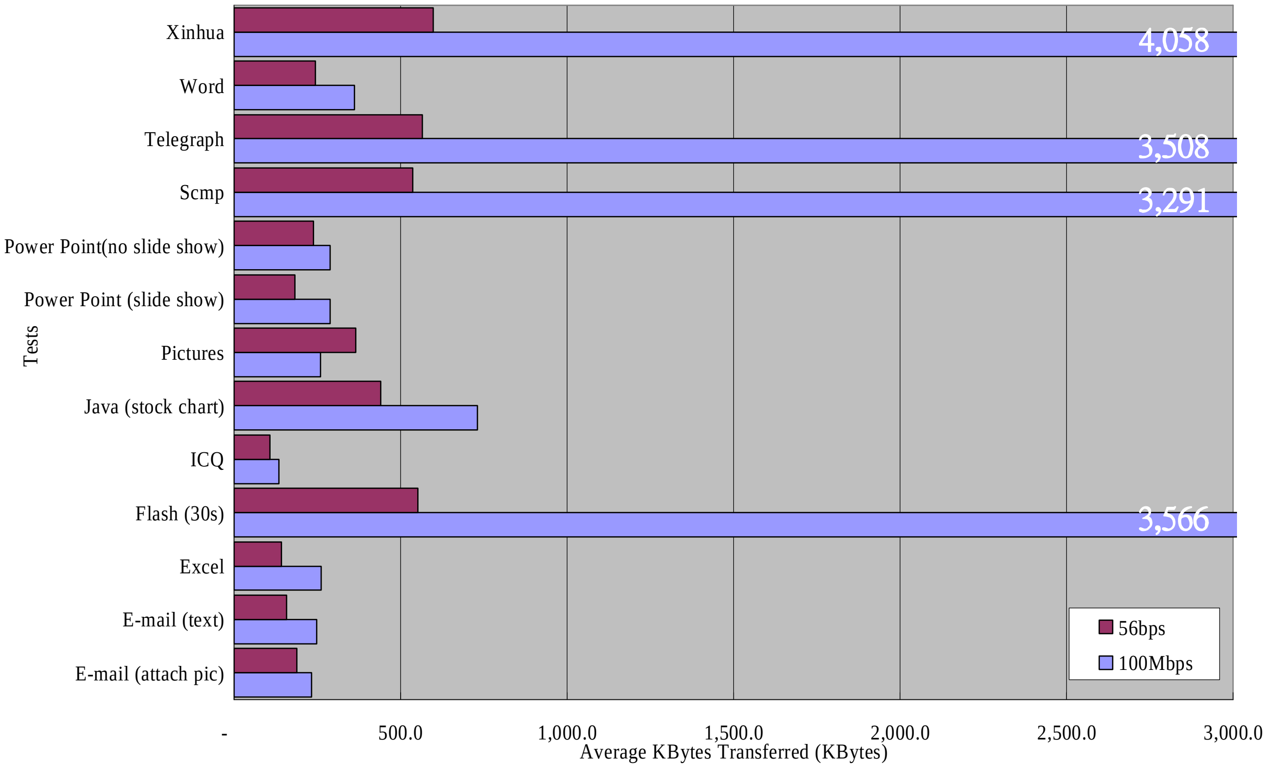


Fig. 12 No. of packets transferred in different tests (100Mbps, 56kbps)

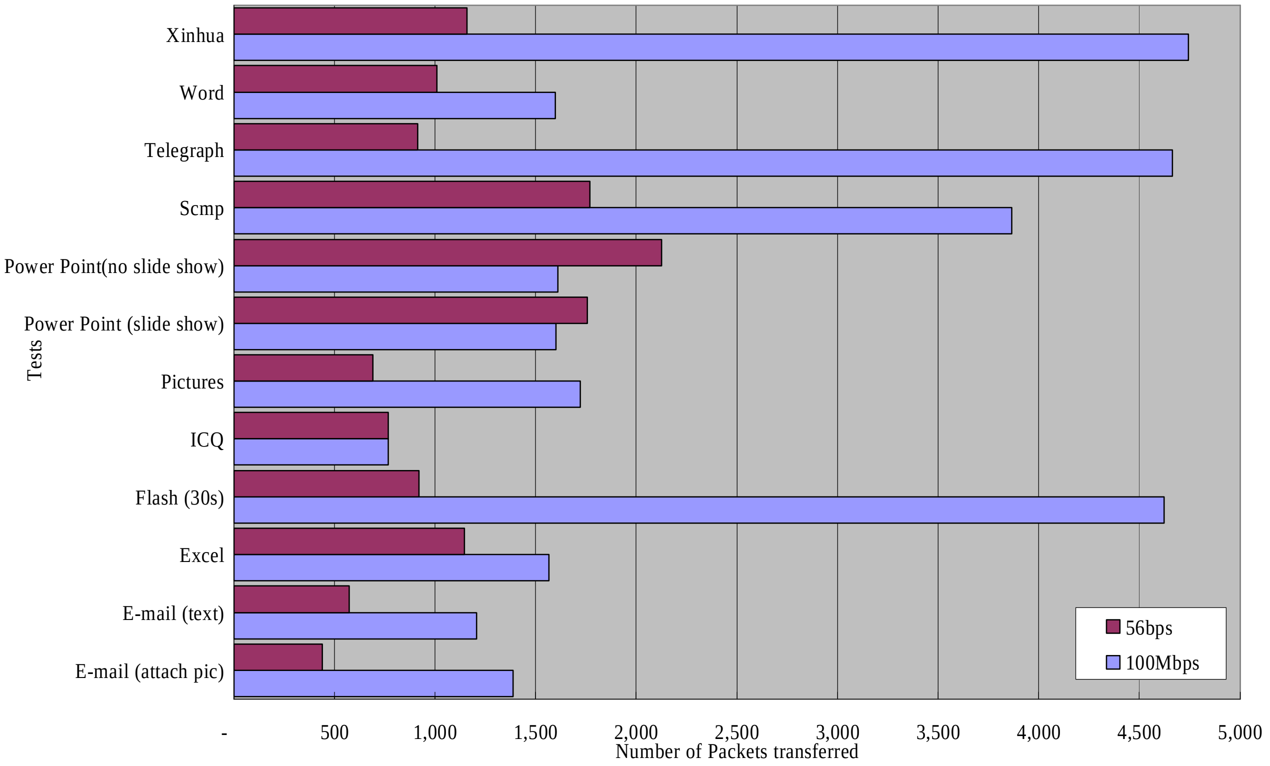


Fig. 13 Time for completion in different tests (100Mbps, 56kbps)

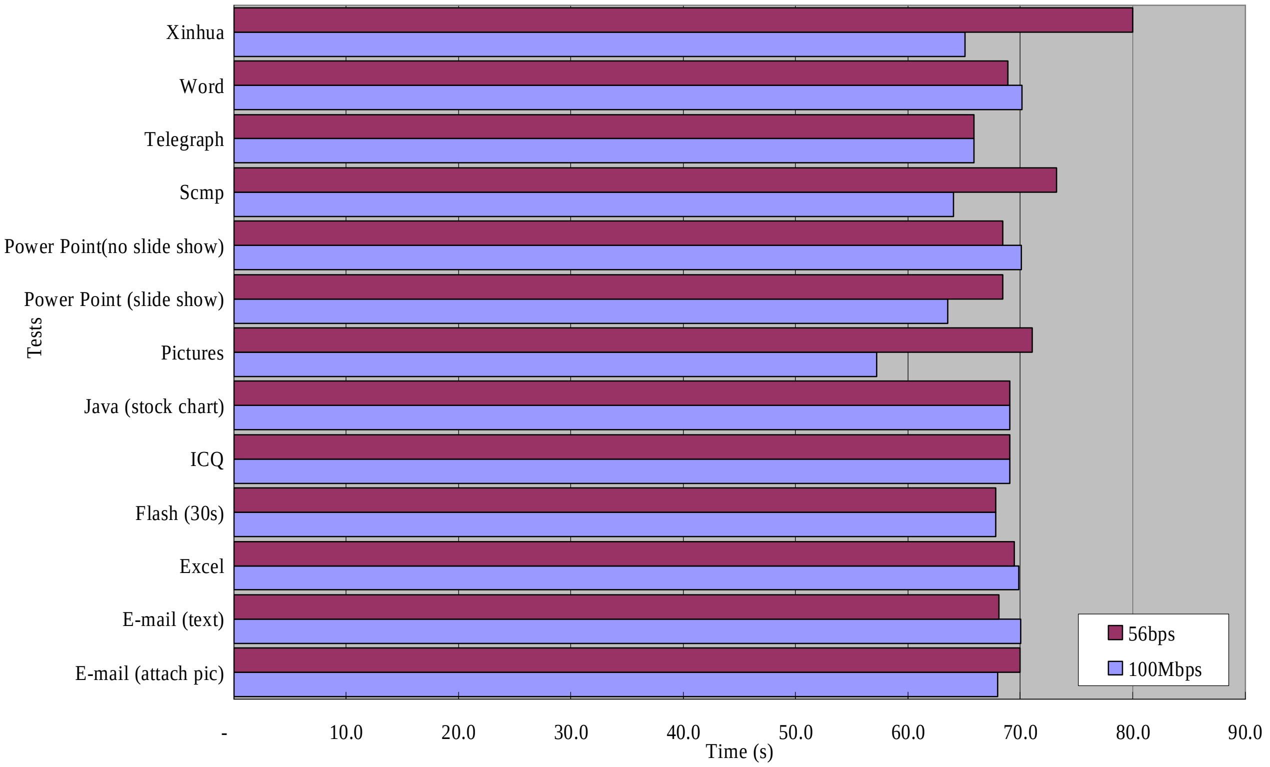


Fig. 14 Kbytes transferred for the same tests during different times (100Mbps, 56kbps)

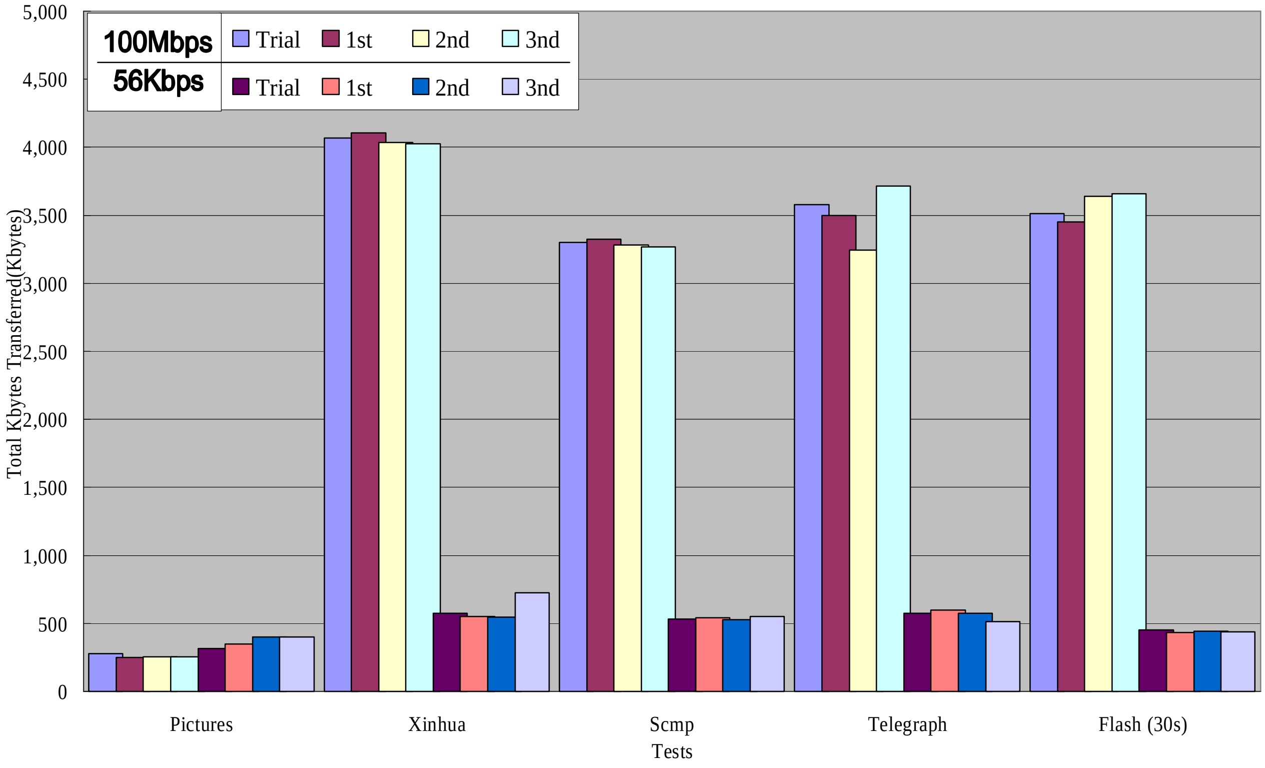
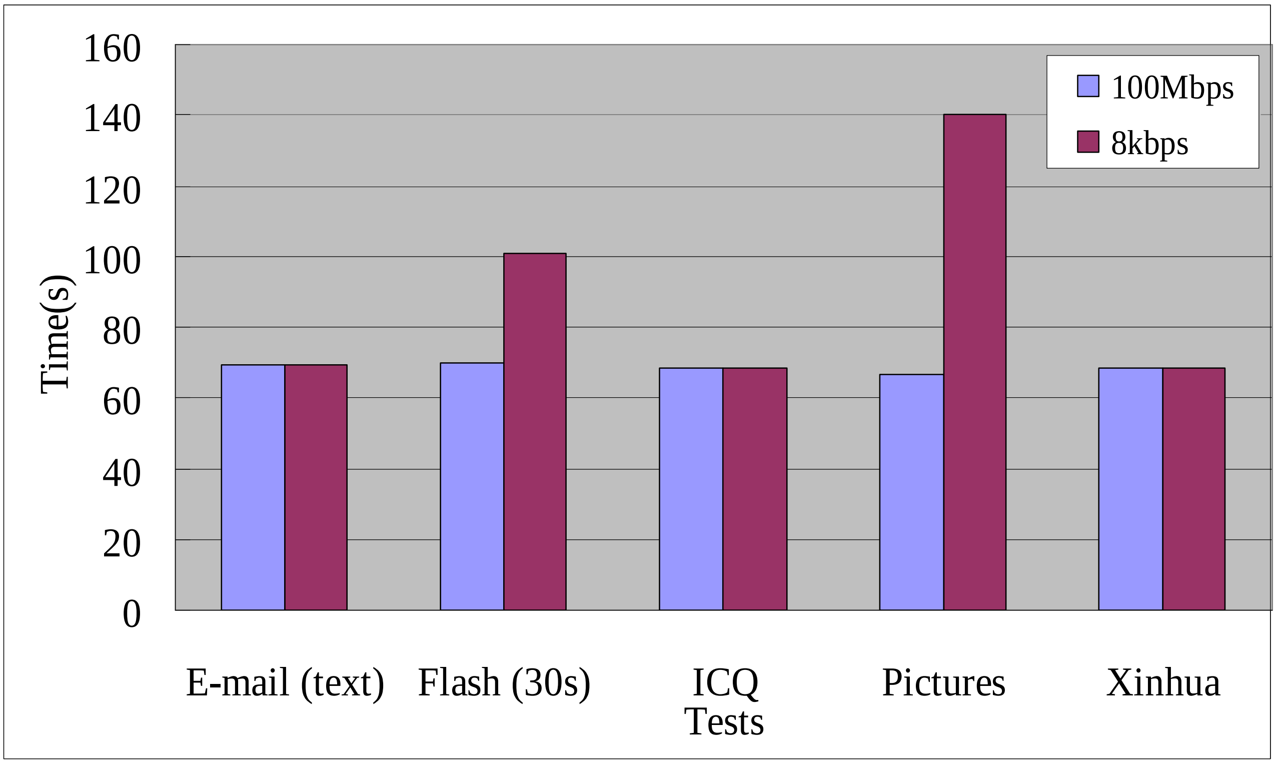


Fig. 15 Time for completion for different tests (10Mbps, 8kbps)



2.4 TightVNC

2.4.1. Introduction

TightVNC is an enhanced version of VNC that includes new features, improvements, optimizations and bugfixes over the original VNC version. TightVNC is fully compatible with the standard RFB protocol used in VNC, so TightVNC viewer can be used with the standard VNC server and vice versa.

Some new features embedded in TightVNC include efficient compression algorithms, configurable compression levels, optional JPEG compression, enhanced web browser access, and so on. [8] Tight encoding is introduced in TightVNC, aimed at optimizing the performance of traditional VNC encodings for slow and medium-speed connections. Together with the standard VNC encodings, TightVNC can operate efficiently over fast networks. Thus, TightVNC claims that people can work remotely almost in real time in most network environments. Any appropriate level of compromise between compression ratios and coding speed can be chosen depending on the connection speed and processor power. JPEG compression can be used to compress color-rich screen areas much more efficiently. TightVNC also includes an improved Java viewer with full support for Tight encoding, local cursor feature, 24-bit color mode, and more. The Java viewer applet can be accessed via built-in HTTP server in the standard VNC.

A lot of improvements, performance optimizations and bugfixes are still carried out for TightVNC. [9]

2.4.2. Tests Result

The result of TightVNC is very similar to that of RealVNC. In this part, we will focus on the difference between these two remote desktop applications.

I. in LAN environment – 10-100Mbps

In LAN environment, there are more applications that start to skip screen updates.

In the cases of Word, Excel, Email and ICQ, the updates observed are quite smooth. Since in TightVNC, CopyRect encoding is enabled, one would expect that the behavior of TightVNC and RealVNC would be similar. CopyRect (copy rectangle) encoding is an encoding inherited from RealVNC, giving “a position in the framebuffer from which the client can copy the rectangle of pixel data.” [10] However, even if CopyRect encoding is used, the amount of data transferred is still the largest among the five remote desktop applications. (Table 4)

In other cases like Powerpoint, Webpages, Picture, Java and Flash, the result is also similar to that of RealVNC. However, the amount of data dropped due to skipping of screen updates is more obvious. For example, in Webpages, the amount of data transferred is smaller than that of RDP, RDPW and Citrix occasionally. For Flash, the amount of data sent is the least among the five remote desktop applications, about 10% of the amount of data sent by RealVNC and about 2% of the amount sent by RDP. (Table 4)

Caching is not effective in TightVNC, as the results show no big differences when the tests are repeated for four times. (Fig. 19) It seems that there is no improvement in this area.

The completion time is a little bit longer, about 5 seconds more than the running time of the macro, it is still acceptable. (Fig. 18)

When taking the data/packet ratio into consideration, we observe that this ratio is the second smallest among the five remote desktop applications. (Table 8) Although the amount of data sent is larger compared with RealVNC, the number of packets sent is much larger than that of RealVNC, resulting in a smaller data/packet ratio. This may be an advantage over RealVNC in the sense that if some packets are corrupted in transit, the amount of data needed to retransmit is smaller as the packet size is smaller.

II. In modem environment – 56kbps

In 56kbps environment, skipping of screen updates almost appears in all tests.

For the cases of Powerpoint, Webpages, Picture, Java and Flash, the skipping of screen updates is much more obvious. As expected, the behavior of TightVNC is similar to that of RealVNC, in the sense that the screen is also updated block by block. In particular, for Flash and Webpages, the drop in the amount of data transferred is larger in TightVNC than in RealVNC; while the opposite occurs in Pictures and Powerpoint.

The same observation as RealVNC appears when the content is changed by scrolling up and down. This suggests that TightVNC uses the same approach when dealing with scrolling a window, i.e., it sends the whole screen again, even though the previous screen may consist of part of the new screen. TightVNC seems to have no improvement in this area over RealVNC.

For Word, Email, Excel and ICQ, the behavior of TightVNC is similar to RealVNC, but with a larger amount of data transferred.

When updating the screen, TightVNC attempts to display the whole previous screen before starting to display the new screen. This is one of the possible reasons to account for the extremely large latency observed in scrolling Webpages, for example, about 156 seconds is needed to complete the macro that would finish in 65 seconds in LAN environment. (Fig. 18) The behavior in scrolling Webpages is not acceptable.

In 56kbps environment, we also observe that the amount of data sent is considerably smaller compared with LAN environment. (Fig. 16) The amount of data sent in 56kbps environment varies from about 20% to about 90% of the amount of data sent in LAN environment. The percentages shown are similar to that of RealVNC.

In 56kbps environment, the data/packet ratio is the smallest among the five remote desktop applications. (Table 8) This is significant in such a low bandwidth environment, as one would expect that more packet loss will be observed in the Internet. If TightVNC is employed in the Internet, the traffic generated due to retransmission of lost or corrupted data would be smaller, and the latency observed may be smaller, compared with the other remote desktop applications.

III. Worst case – 8kbps

In this extremely low bandwidth environment, we expect that the performance of TightVNC would be even worse than that in 56kbps environment.

In the test cases like Email, Picture, Flash and Webpage, the amount of data sent drops to less than 25% of the amount of data that would be sent in LAN environment. (Table 11, 12) The possible reason is that these test cases consist of lots of color variations or screen updates due to scrolling, and when the bandwidth available is so low, much of the screen updates would be discarded, resulting in a large drop in the amount of data sent.

In ICQ, the amount of data sent drops to about 60% of the amount of data that would be sent in LAN environment. (Table 11, 12) The possible reason is that the amount of data sent is smaller compared with the tests mentioned above. So it is not necessary to drop the amount of data sent to such a large extent as in the cases mentioned above.

We observe that the time of completion for TightVNC also increased dramatically, except for Email, but the performance is a little bit better than RealVNC, except for Flash. (Fig. 20) The time of completion observed is less than double of that observed in LAN environment in general; and in Flash, the time of completion increases to five times that observed in LAN environment. Still, the performance in 8kbps is not acceptable in general.

IV. Problem of leakage

The same problem of leakage occurs in TightVNC. Even when the server/client is idle after completion of macro, there are still data transferring from the server to the client. The amount of data transmission caused by leakage is rather constant for each test case and link bandwidth. The amount of data sent due to leakage is about 1.5 kilobytes for every 20 seconds.

V. Different settings in compression and quality

This part is unique to TightVNC, as RealVNC does not have the options to manually configure the compression level and quality setting.

The test we performed is viewing Pictures in LAN environment. The compression levels chosen are 1 (fast compression), 5, 9 (best compression); while the quality settings used are 0 (worst quality), 5, 9 (best quality).

The result is quite intuitive. For the same quality setting, the amount of data transferred is smaller if a larger compression level is chosen, which corresponds to higher compression ratio. For the same compression level, the amount of data transferred is smaller if the quality setting is smaller, which corresponds to worse image quality and less information. (Table 1)

We observe that the impact of quality setting on the result is smaller than that of the compression level. This may be due to the fact that the test is carried out using 8-bit display, which can be viewed as using a low quality setting, and some information is lost. The screen updates are already compressed by using 8-bit display to an extent such that the quality setting has little effect on the amount of data sent. If the test is carried out with 8-bit display disabled, we would expect that the quality setting has a more significant effect on the amount of data transferred.

Table 1 Comparisons on different compression and quality setting.

		Std Dev			Average Byte Transfer				
		Quality Setting							
		0	5	9	0	5	9		
Compression	1	32,956	17,418	4,989	574,420	567,403	599,061	1	Compression
	5	1,505	2,435	4,078	297,390	301,929	309,632	5	
	9	5,500	4,576	4,151	282,268	288,802	296,339	9	
	1	0.0%	-1.2%	4.3%					
	5	-48.2%	-47.4%	-46.1%					
	9	-50.9%	-49.7%	-48.4%					
		0	5	9					
		Quality Setting							
		Reduction Ratio							

IV. Charts

Fig. 16 Average Bytes transferred in different tests (100Mbps, 56kbps)

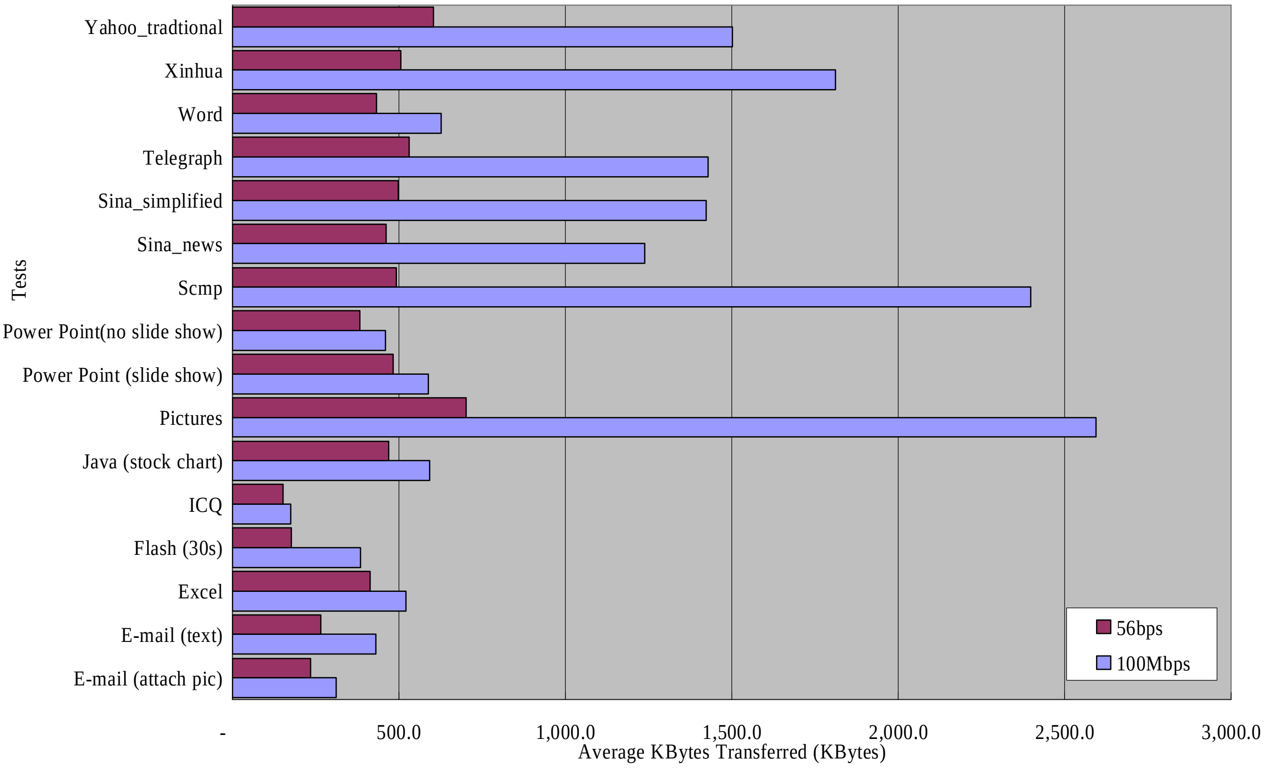


Fig. 17 Number of packets transferred in different tests (100Mbps, 56kbps)

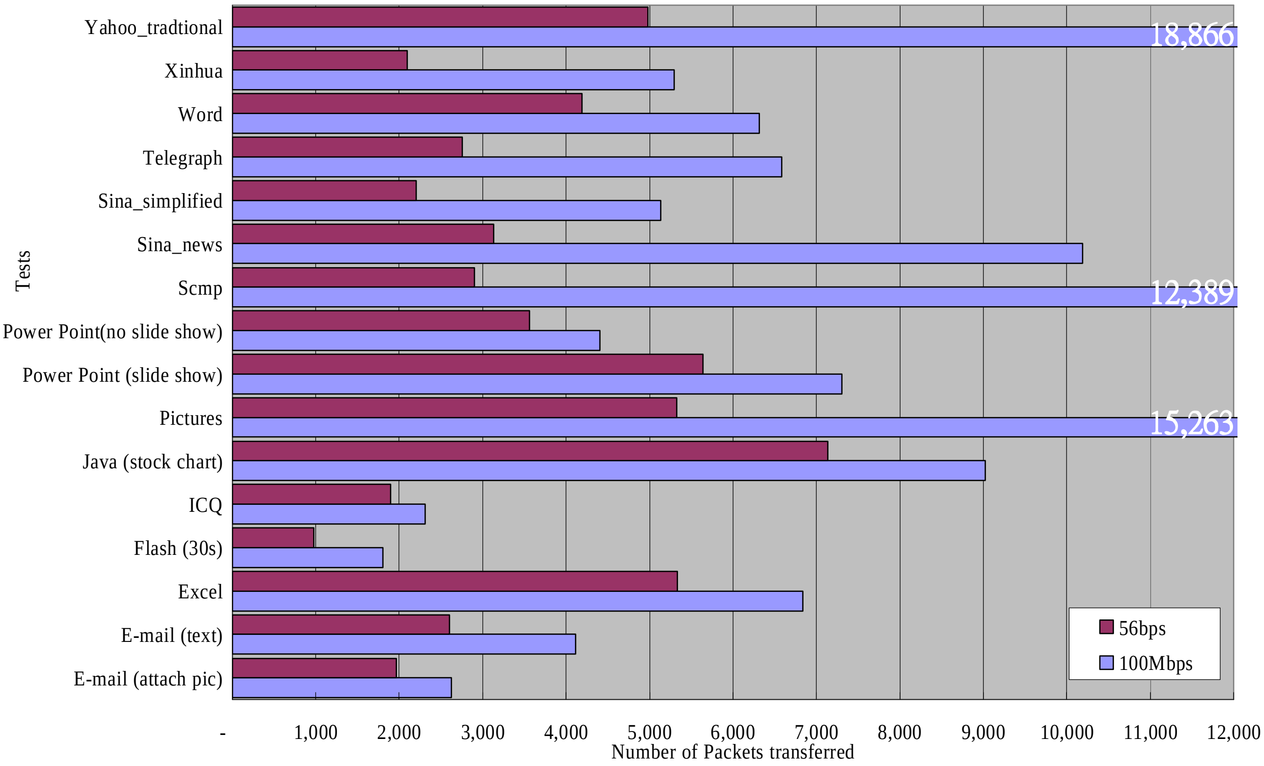


Fig. 18 Time for completion in different tests (100Mbps, 56kbps)

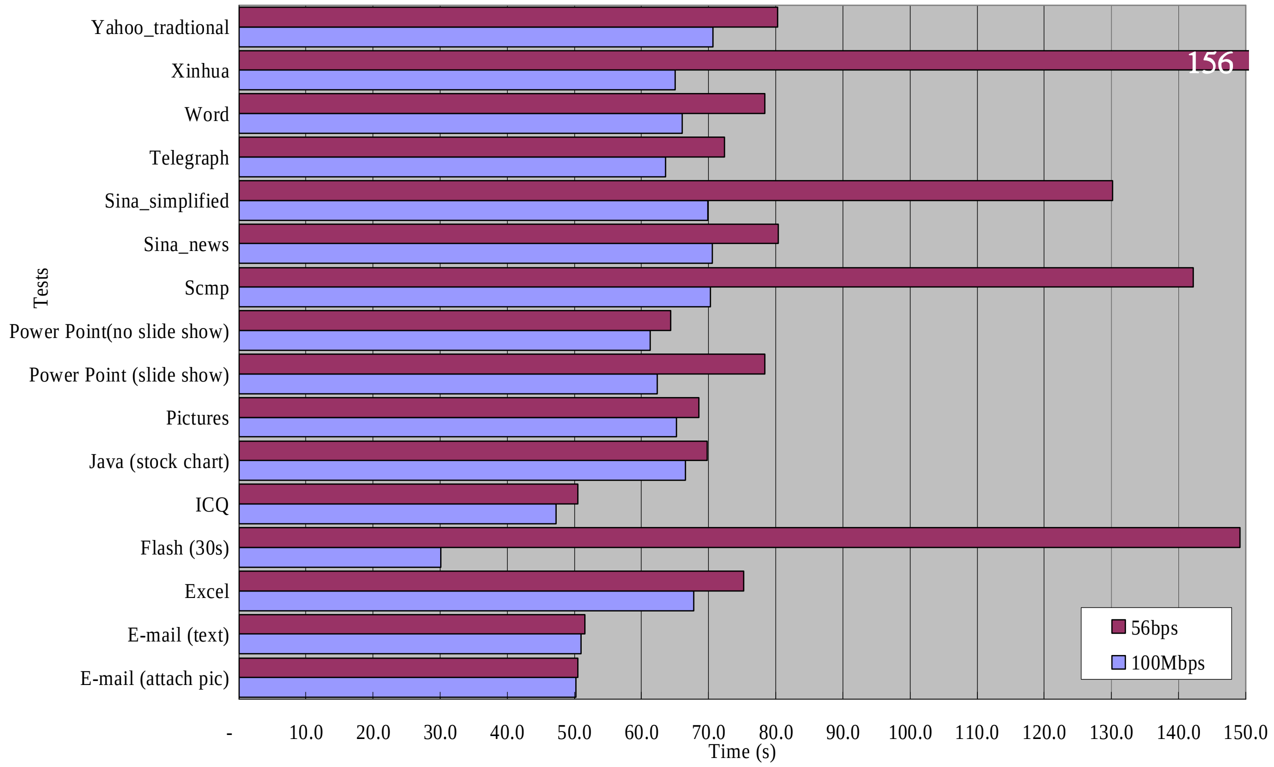


Fig. 19 Kbytes transferred for the same tests during different times (100Mbps, 56kbps)

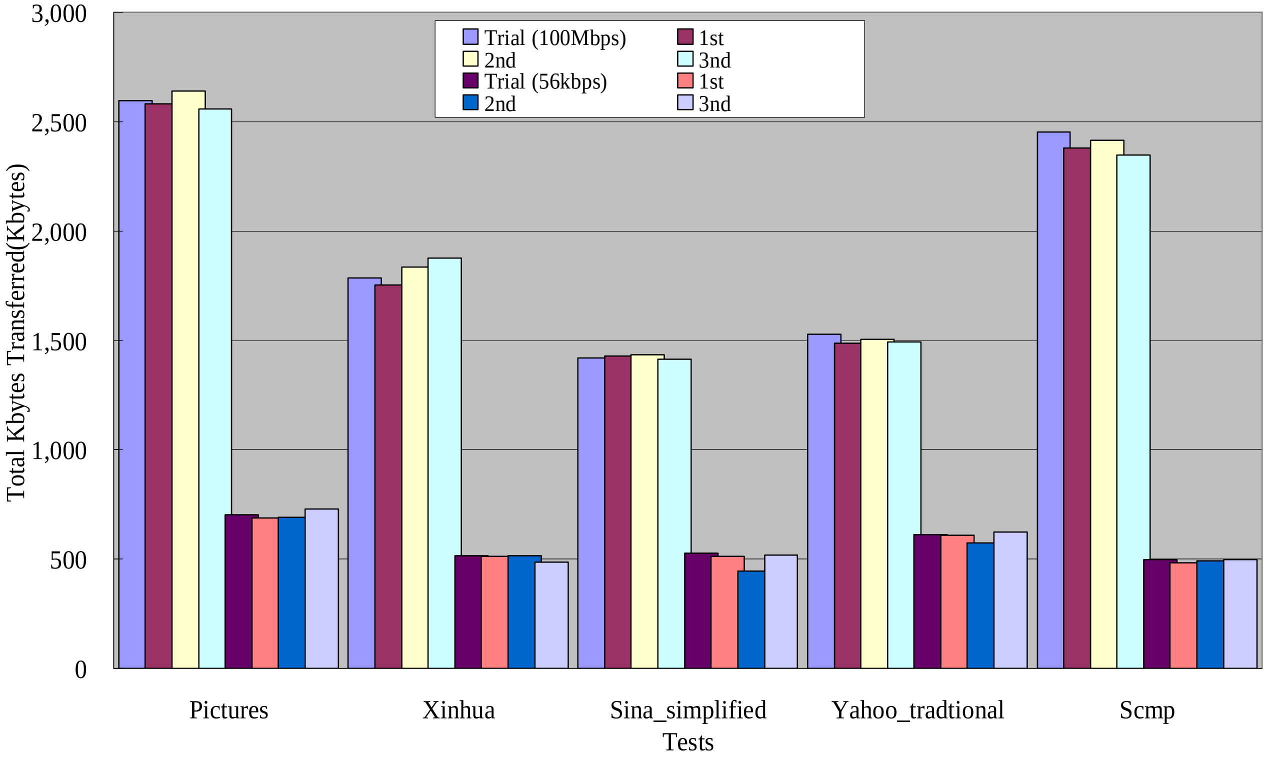


Fig. 20 Time for completion for different tests (10Mbps, 8kbps)

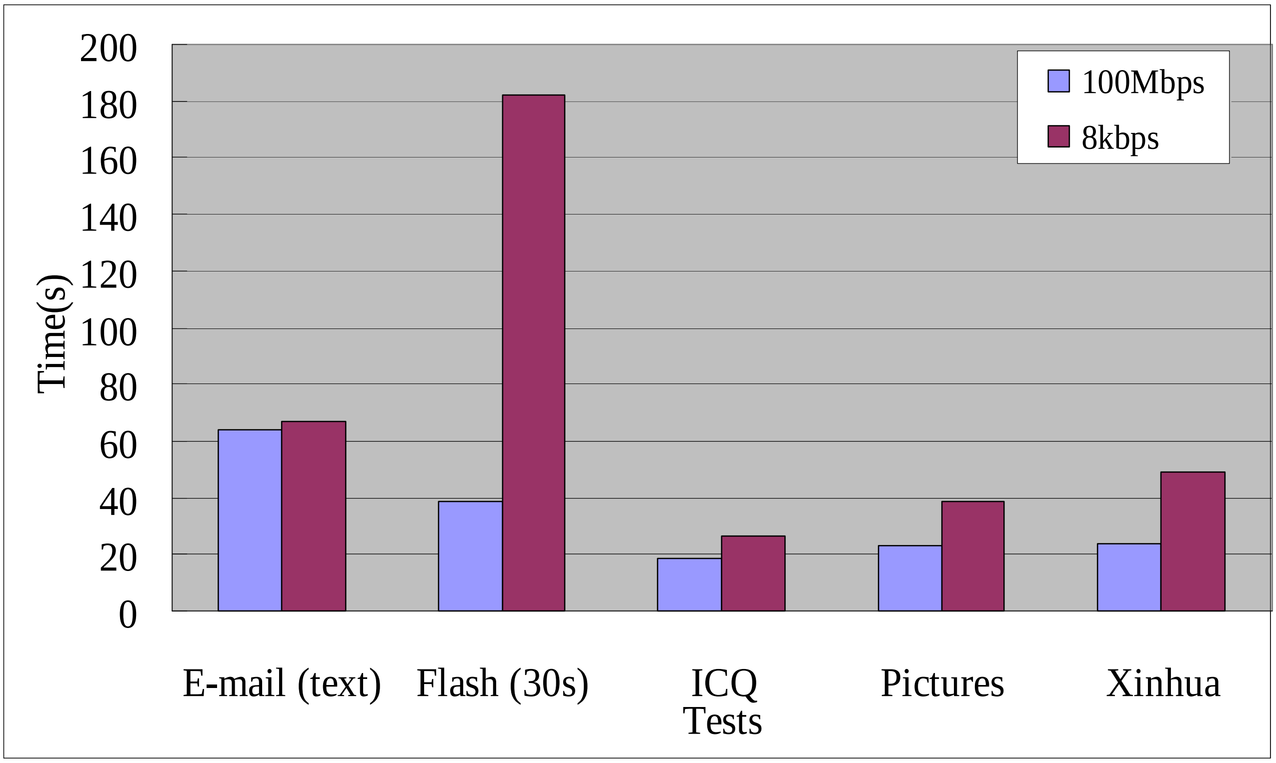
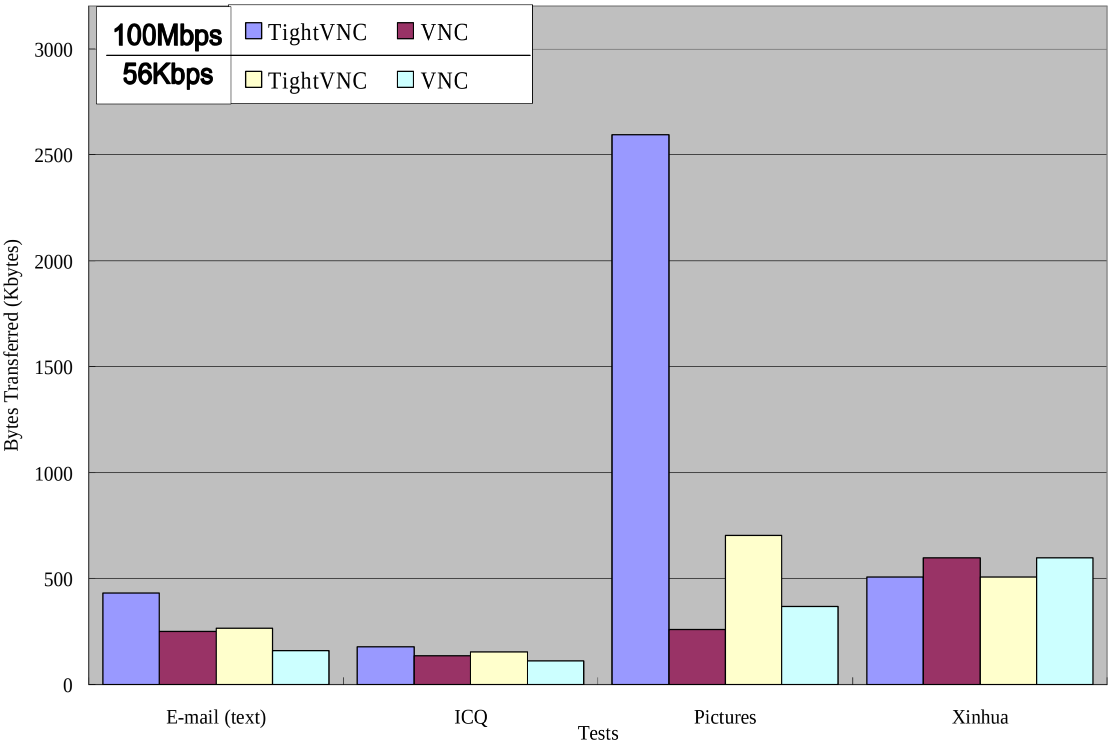


Fig. 21 Comparison with VNC in byte transferred of different test and different environment



2.5 Citrix

2.5.1. Introduction

Citrix Systems, Inc. is a global leader in application server software and services that offer Digital Independence, the ability to run any application on any device over any connection, wireless to Web. While the Citrix technology enables organizations to provide access to server-based applications from a wide variety of client devices and platforms. While the cross platform behavior are supported with the help of Citrix ICA Client. Since these applications are installed, updated and maintained on central servers instead of each client, the cost and complexity of administration are reduced significantly. Like other Remote Desktop Protocol, the applications execute 100 percent on the server, and only keystrokes, mouse controls and screen updates are transmitted over the network between the server and client. [11]

Citrix enables high application performance even over bandwidth-constrained connections as well as greater data security. Since in this test, we are comparing the performance on low bandwidth environment, we don't apply any encryption method in the server. Since Citrix Systems is founded in 1989, it is staying as a leader in RDP field. Even the Remote Desktop in Windows XP comes from their technology. This brings a very much interest to observe how it is superior to the other.

2. Tests Result

Before we discuss the tests results and the observation we would like to observe the relationship between the data:

The no. of packets transferred is proportional to the average bytes transferred. For example, in email (text) test, even though in different environment, the ratio of no. of packets transferred to average bytes transferred is more or less constant (Table 2). This relation holds when we understand the TCP data packet's structure. Moreover, follows from that relationship, the ratio of TCP packets (data + control) and data packets are proportional to the total bytes transferred (data + control) and data byte transferred. For example in email (text) test, both ratio ranges from 1.2 to 1.4 and 1.5 to 1.8 respectively, for which they are similar (Table 3). A last remark to be noted on the above comment is that the ratio rise when the bandwidth is become lower. It happens because the bandwidth is not able to transmit a large packet out.

Consider the result of data bytes. Since the Citrix ICA Clients are doing a magnificent job in caching, we can easily verify this job from the continuous tries on the same test. Besides, its behavior on the caching of Picture and the text, will be discussed later.

Table 2

E-mail (text)	Average total bytes transferred (Kbytes)	Average packets	Average Kbytes per packet
100Mbps	42.6	367	1.15
56Kbps	70.6	604	1.16
8Kbps	37.0	234	1.58

Table 3

E-mail (text)	A. Average total bytes transferred (Kbytes)	B. Average data bytes transferred (Kbytes)	C. Total packets	D. Data packets	A to B ratio	C to D ratio
100Mbps	70.9	58.7	554	350	1.21	1.58
56Kbps	71.5	59.5	604	361	1.20	1.67
8Kbps	39.1	27.9	420	234	1.40	1.79

Now, let us come back and observe the result in the LAN environment, modem environment, and the worst case we proposed (8kbps)

I. in LAN environment – 10-100Mbps

Generally, in LAN environment (100Mbps), all remote desktop should have performed very well. And this assumption still holds on Citrix, all the application run smoothly as if you are using your own computer. In order to discriminate the better from the worse, we needs further understanding upon the byte transferred sent through the network. First of all, a general understanding on the average bytes transferred is proportional to the frequency of screen updates. And the processes of converting the screen we see back to data have already been mentioned in RDP part already.

For those test that simply relies on typing and scrolling of text only, like Word, Email (text), ICQ, the resulting bytes transferred is very small (Fig. 22). And for those tests that relies on showing and scrolling of picture, like those Web Page tests (Flash, Pictures, etc.), the resulting bytes transferred become at least a multiply of 20 to the text-based test. Though in LAN environment, user will not observe any difference in latency in difference on the screen. If the users are performing file transfer process, however, glitches or latency would have occurred. So after a small test with this idea, except the Web Page (Flash) test, all other test does not have any problem from user's point of view.

II. In modem environment – 56kbps

Up to this stage, this test starts to provide more observation. Precisely, except Web Pages test (including flash, Xinhua, etc), all other test still performed as same as before. (This can be easily verified by observing the difference of the data byte transferred in two conditions in Fig 22).

For the first trial of the Web Page test (except flash), time is needed to load the picture. However, since we have introduced delay between each input, and Citrix still can barely enough to catch up. For example we have introduced some small breaks between pressing "page down", the small breaks gives enough time for the client to receive enough information shown to users. And for the user's point of view, this appearance looks like they are using a slow computer to browse homepage. Moreover, the showing pattern of the Web Page is showing the page completely at the same time. This pattern is quite interesting when comparing with other software, as it means that Citrix can decide whether it is better to wait for receiving whole information before showing up.

Now, let us observe the flash test. This test should have consumes most of the bandwidth, and revealed the properties of different RDP in dealing with congestion. In this test, it has shown Citrix is good in maintaining the system clock (Fig 27, the time is more or less constant), however but skipped most information (254kb). During the playback of the flash movie, the clip is shown patch by patch only, without showing the same frame at the same time. The latency is reduced with this method. Though the user cannot view the clip normally, the users can still catch up what the computer has shown by those patches. And this patch by patch showing scheme is different to the Web Page test (which shown up totally together). This is the case where the bandwidth is not enough to reflect the information shown on the screen.

Overall to say, the average bytes transferred this time is much smaller to the 100Mbps case even the same tests are carried out (Fig 22). As this is the bandwidth reduction features of the Citrix, which supports various mechanisms to reduce the amount of data transmitted over different network connection. Sophisticated algorithms are carried out to determine when to send screen updates to the client. This enables a better performance in lower bandwidth, while the users observe barely difference on the screen.

Last of all, the caching effect is shown in the low bandwidth environment (Fig 25). The bytes transferred decreases less significantly than in the 100Mbps case. This is because Citrix have detected the bandwidth is enough without stronger compression, and so leave it alone in order to save the CPU resources. Meanwhile, the Webpage (Xinhua) test in 56kbps environment has shown clearly a general downward trend. Consider the SCMP test (fig 26), in both environments, the bytes transferred fall continuously and approaches to limit. This is due to the fact that the compression and caching has already reached its bound. However, in the 6th test, the byte transferred have increased back again, this is because of the flushing of cache. Besides the bytes transferred, the latency is reduced also. Consider the 56kbps environment, the latency on the web page during the trial disappears totally when running the test second time. This has shown that the ICA Client's caching effect is powerful when comparing to RDP and TightVNC.

III. Worst case –8kbps

Last of all is the worst case we proposed. This test is constructed with the shed of light of GSM mobile phone system. While we aimed on several test which are more significant to the others. Since Citrix has a promise of quality already with bandwidth more than 160Kbps, now we fulfill this condition with twentieth only. This brings a harsh environment to all remote desktop applications, as well as giving a good observation on the environment operated in mobile systems.

Unlike RDP and RDPW, the connection in Citrix will not drop out due to insufficient bandwidth. And the mouse movement will not have latency as other software. This performance can be further perfect by the "SpeedScreen Latency reduction Manager" from the MetaFrame Server. Though in this test, we do not turn on specialized options to improve Citrix's performance, the latency in mouse still undetectable from user's point of view.

Considering the test, we can split the observation into three parts:

Firstly, the flash, which can be described as dynamic picture. The performance of it is totally unbearable. While comparing to the modem environment, the appearance of patch become slower and smaller in size. And the latency is large (Fig 27), but still performs best among them.

Secondly, the test, which are consist of scrolling and typing of text (ICQ, etc.) The performance of it is very good, as if no difference from the LAN and modem environments. When we observe the data, it can easily know that such a small data transferred will not make any difference. And this brings an idea that Citrix used certain caching method to send the text. However, we can compare that to seeing simplified Chinese site (Xinhua) (Fig. 22), the performance become less desirable, as Citrix is developed in US, and those Chinese may be treated as picture and makes such a difference in the webpage tests. Meanwhile the conclusion on Citrix is performing well and stable in text-based application.

Last of all is the other Web Page test. On the first trial, the latency of seeing a new page and scrolling is quite big. However, when we do the test again after the trial, the test performs very smooth. Citrix's cache has cached the entire page into buffer, and the server commands the client to read back when needed. And we have mentioned in modem environment part already.

After performing these tests, in Fig. 28, it is an abstract on the bytes transferred. A question is raised after the benchmarking, "Can Remote Desktop Applications run on wireless device?" Surely if you are using a pocket pc and want to access your computer at home, it is barely efficient. But this problem is an open problem. For example we can either turns the Pocket PC into console mode which can remote control your computer totally, etc. It will be discuss on the conclusions.

IV. Charts

Fig. 22 Average Bytes transferred in different tests (100Mbps, 56kbps)

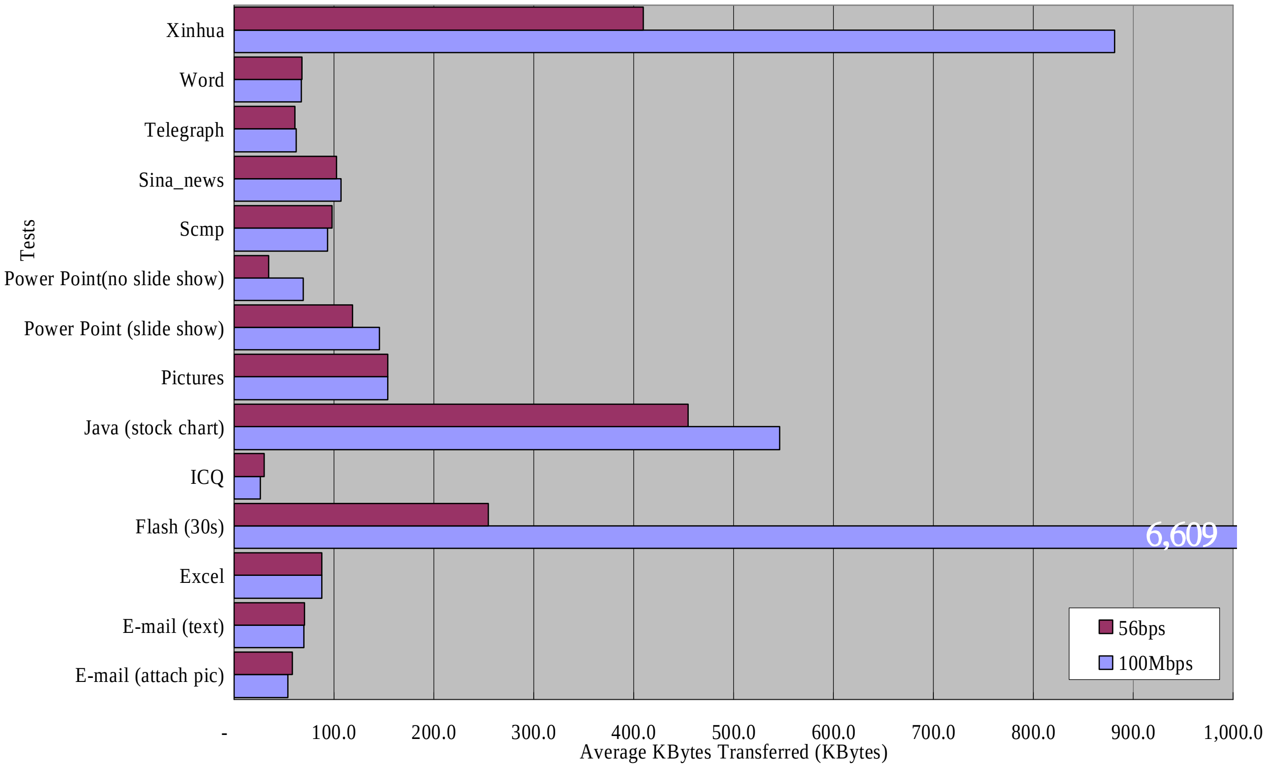


Fig. 23 Number of packets transferred in different tests (100Mbps, 56kbps)

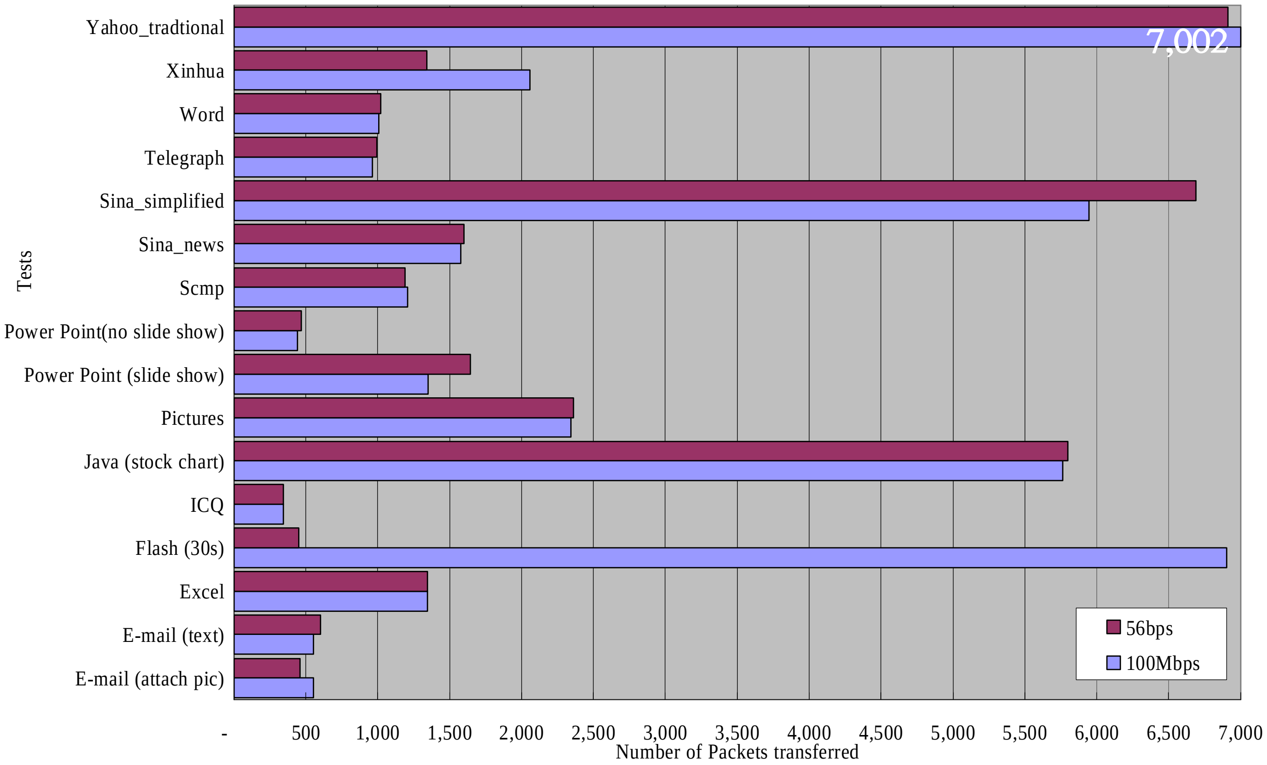


Fig. 24 Time for completion in different tests (100Mbps, 56kbps)

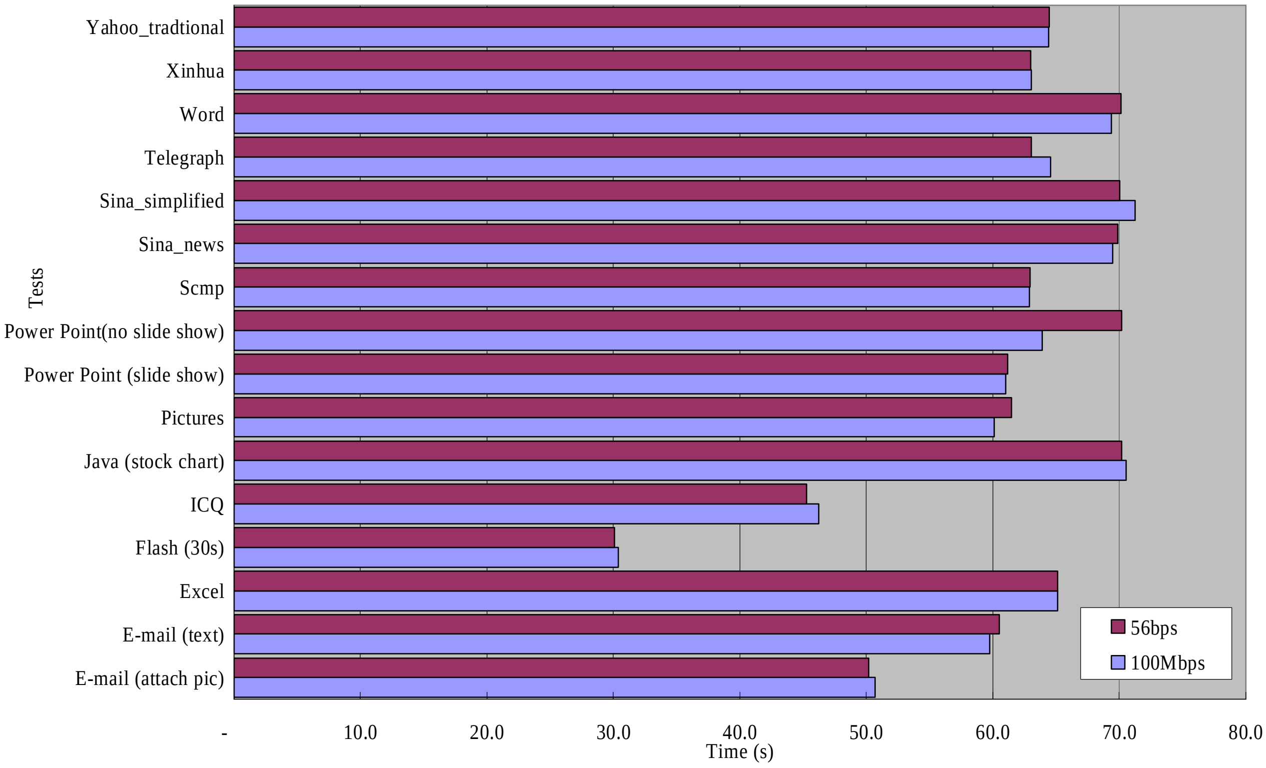


Fig. 25 Kbytes transferred for the same tests during different times (100Mbps, 56kbps)

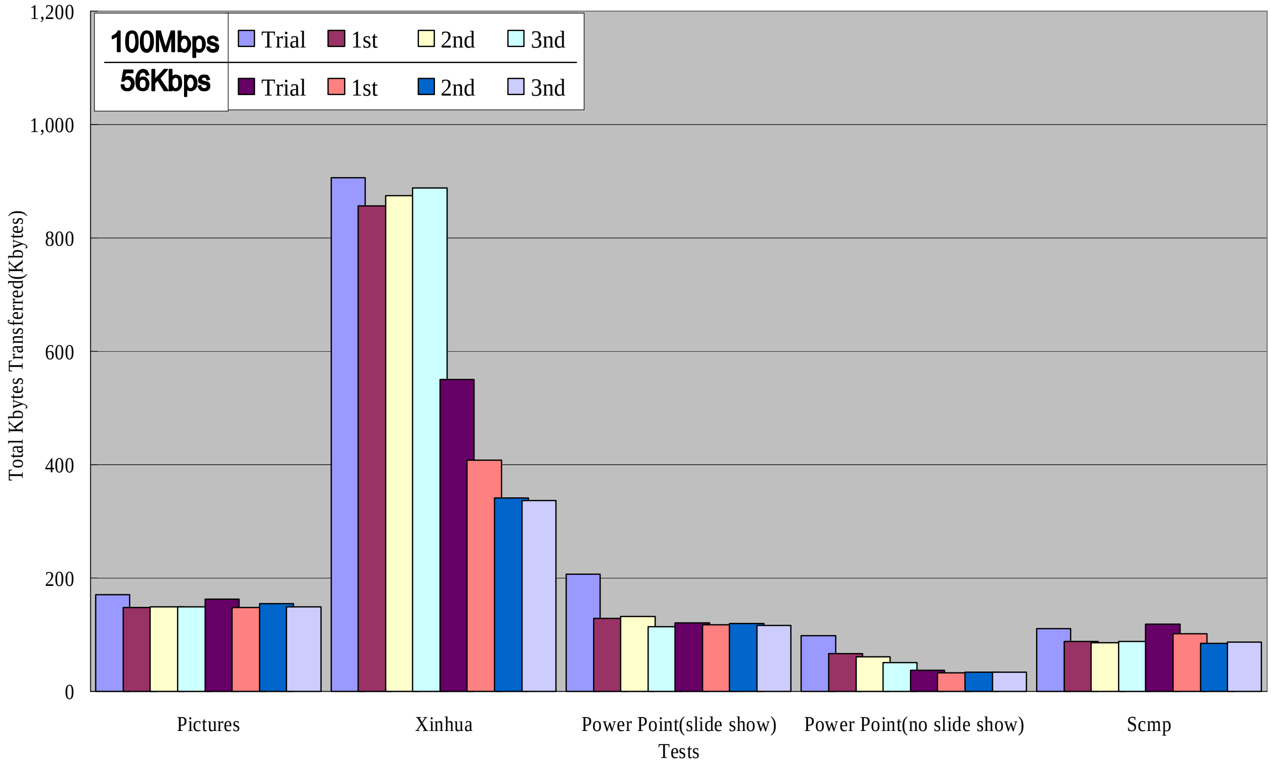


Fig 26 Continuous performance on SCMP test in different environment (100Mbps, 56kbps)

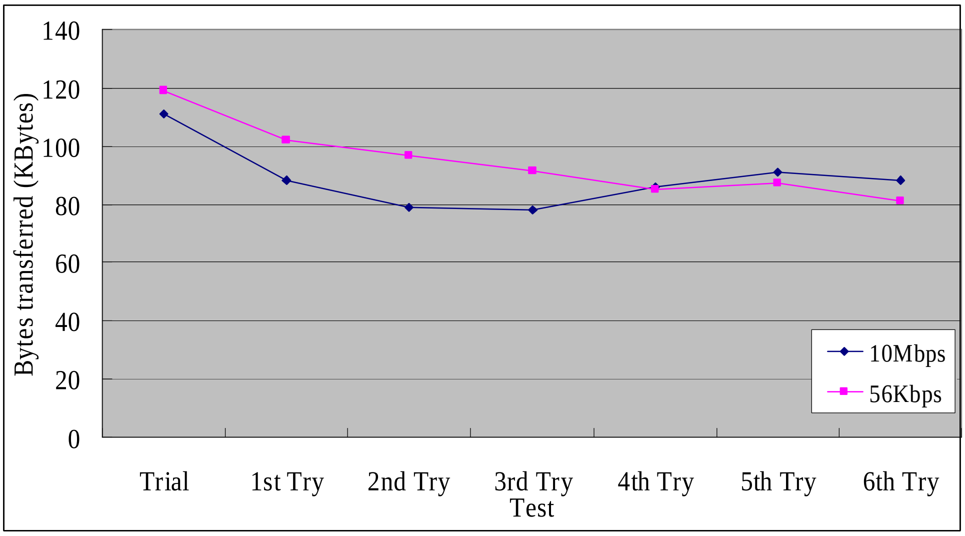


Fig. 27 Time for completion in different tests (10Mbps, 8kbps)

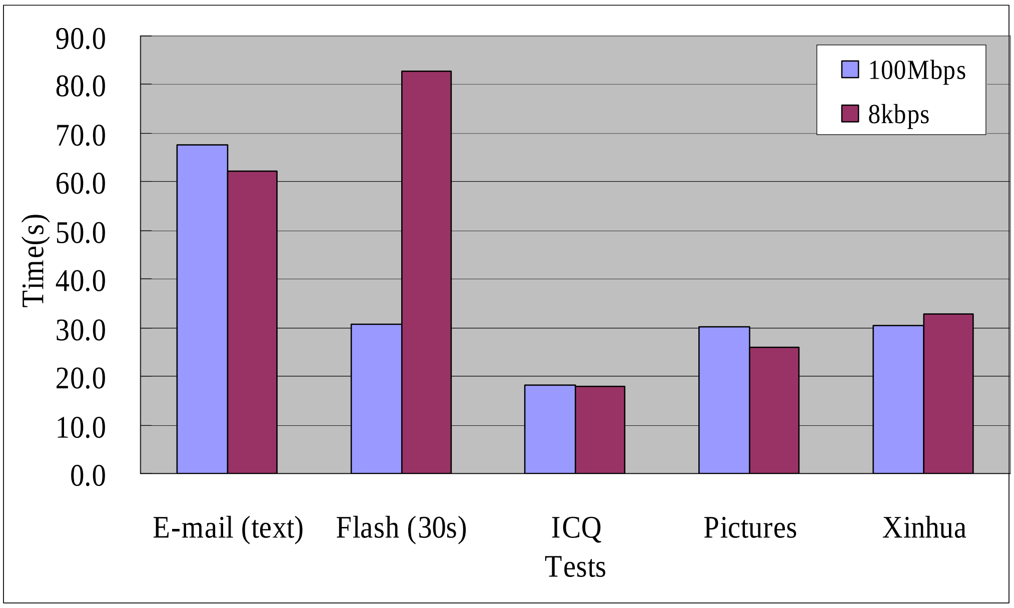
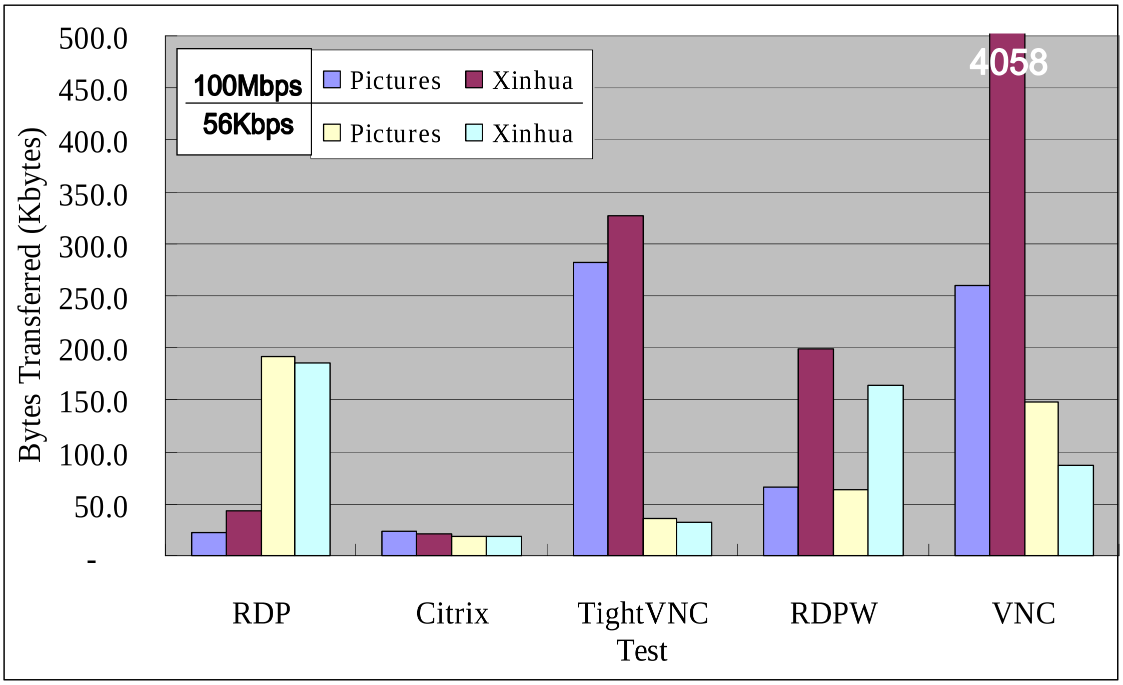


Fig 28 Cross comparison in different tests on bytes transferred



Session 3
Conclusion

3.1 Summary of benchmarking

Overall, our result indicates that all remote desktop applications perform well in LAN (100Mbps/10Mbps) environment. Daily tasks such as word processing, E-mail, ICQ, web pages browsing etc all run smoothly. A little bit exception for the case in Flash, in which discontinuity is observed in all applications – it is severe in RealVNC and TightVNC.

However when the bandwidth drops (56kbps/8kbps), the performances for different applications vary significantly. For RealVNC and TightVNC, the latency for scrolling, keystroke and mouse movement is significant and this is true for all tests. They will skip screen updates when they cannot keep up with the server and this is the most serious in the Flash test. This implies multimedia intensive applications will not be possible for RealVNC and TightVNC in such low bandwidth environment.

In RDP, the performance in text-based applications (Word, Excel) is still satisfactory and latency is not significant. This is due to the caching of glyphs and fragments mechanism. Yet in those tests which make extensive use of bitmaps (in web pages tests, ICQ, Flash etc., the data sent from the server are mainly bitmap), the performance gets worse. Instead of skipping screen updates like RealVNC/TightVNC, it will output all of them which make latency more obvious – and it is the most severe in the Flash test. However things go better as the tests are repeated. This is due to the persistent bitmap caching property (but this is not true in Flash, since there are too many screen updates and the cache will overflow). Upon the case of too many lost packets, RDP will simply disconnect.

The performance for Citrix is the most satisfactory. The text-based applications run smoothly all the way. For the other tests, since it also has the bitmap caching property, the latency is shorter than in RealVNC and TightVNC. On the other hand, it will skip some screen updates in order to keep up with the server, which further shortens the latency. In very low bandwidth environment, it will still try hard to continue its job rather than disconnect like RDP.

In summary, factors which affect performance are:

- Caching - bitmaps, glyphs and fragments
- Intelligently identify the type of data sent – whether it is text or bitmap, as sending glyphs rather than bitmaps can lower the data size.
- Intelligently identify the input – if it is scrolling, the data size will be smaller for moving the content up/down, rather than sending all the data once again to the client.
- Skip screen updates in an optimal manner
- Compression changes dynamically upon bandwidth changes

This table shows whether the applications include the above factors:

	RDP	RDPW	Citrix	VNC	TightVNC
Caching	Y	Y	Y	N	N
Identify data sent	Y	Y	Y	N	N
Identify input	Y	Y	Y	N	N
Skip screen updates	N	N	Y	Y	Y
Compression changes with bandwidth	Y	Y	Y	N	Y

This table shows how the applications perform:

	RDP	RDPW	Citrix	VNC	TightVNC
Bytes transferred (1 = least bytes transferred)	3	2	1	4	5
No. of packets transferred (1 = least no. of packets transferred)	3	2	1	4	5
Latency (1 = least latency)	2	3	1	5	4

2. Can remote desktop applications run on wireless devices?

It depends on different networks. The performance is not acceptable in a GSM/2.5G network. The maximum data rate in a 2.5G environment is 50kbps only. As shown in our benchmarking, only RDP and Citrix will barely survive in text applications for such environment. We expect the performance will be even worse in a wireless environment which loss packets are common.

The performance is more satisfactory in a 3G network, in which the data rate ranges from 384kbps - 2Mbps. However, it will be not economically to employ RealVNC and TightVNC. As RealVNC and TightVNC will send much more data than RDP and Citrix in the same test, the users will need to pay more. This is because the users pay according to how much bandwidth they use.

More things can be done to improve the performance, like caching, compressions etc. At the same time, with the advancement in mobile hardware technology, our handheld devices are becoming more powerful. A good example will be the Intel PXA27x family of processors. It can handle multiple forms of wireless broadband access with enough computing power to provide cell phones with full motion video conferencing capabilities and PDAs with DVD-quality video playback. With the enhancement in the applications, network and hardware, this design – remote desktop applications run on wireless devices, will likely be possible in the future.

Session 4

Appendix

4.1. Data

Table 4 Average Byte Transferred in different environment (10Mbps & 56Kbps)

			Average (Kb)				
	Test Case	Orig Data Size (Kb)	RDP	Citrix	TightVNC	RDPW	VNC
Bandwidth: 10Mbps	E-mail (attach pic)	15.0	77.7	54.0	312.7	94.5	232.1
	E-mail (text)	11.0	87.0	70.1	431.8	90.5	249.5
	Excel	-	90.9	87.7	520.7	87.2	261.5
	Flash (30s)	-	17,992.4	6,609.5	385.1	10,437.5	3,566.3
	ICQ	-	98.8	26.8	176.3	65.8	135.8
	Java (stock chart)	-	1,607.3	546.5	593.4	1,860.2	730.6
	Pictures	216.0	350.2	154.2	2,593.1	325.5	260.2
	Power Point(slide show)	-	328.7	145.6	588.3	413.8	288.7
	Power Point(no slide show)	750.0	299.2	69.3	460.1	-	288.7
	Power Point(slide show)	-	328.7	145.6	588.3	413.8	288.7
	Scmp	266.0	2,007.3	93.4	2,398.2	1,755.1	3,291.8
	Sina_news	72.1	99.4	107.2	1,239.2	-	-
	Sina_simplified	533.0	1,659.0	4,681.8	1,423.4	1,659.0	-
	Telegraph	121.6	1,178.4	62.6	1,429.5	-	3,508.6
	Word	-	62.7	67.5	627.5	60.1	361.6
	Xinhua	315.0	2,412.2	881.5	1,812.0	3,107.7	4,058.4
	Yahoo_tradtional	176.0	796.2	3,915.9	1,502.4	-	-

			Average (Kb)				
	Test Case	Orig Data Size (Kb)	RDP	Citrix	TightVNC	RDPW	VNC
Bandwidth : 56Kbps	E-mail (attach pic)	15.0	77.5	58.3	234.3	66.0	189.2
	E-mail (text)	11.0	82.0	70.6	266.4	104.6	158.5
	Excel	21.0	83.3	88.1	413.7	74.3	143.2
	Flash (30s)	-	8,437.3	254.5	178.0	4,506.8	552.7
	ICQ	-	62.4	30.4	152.6	81.1	109.5
	Java (stock chart)	-	530.7	454.5	470.5	625.7	440.8
	Pictures	216.0	141.1	153.9	701.8	109.3	366.1
	Power Point (slide show)	750.0	351.1	118.9	482.4	277.4	183.7
	Power Point(no slide show)	750.0	287.8	34.8	384.2		238.9
	Scmp	266.0	605.8	98.4	492.4	605.8	537.6
	Sina_news	72.1	128.4	102.9	461.5	-	-
	Sina_simplified	533.0	494.4	5,261.6	499.6	-	-
	Telegraph	121.6	245.1	61.2	530.6	-	566.3
	Word	363.0	67.2	68.4	432.4	55.9	244.9
	Xinhua	315.0	729.2	409.3	506.5	1,481.7	598.4
	Yahoo_traditional	176.0	304.0	3,920.0	604.1	-	-

Table 5 Total Packets in different environment (10Mbps & 56Kbps)

	Total Packets									
	Bandwidth: 10Mbps					Bandwidth: 56Kbps				
Test Case	RDP	Citrix	TightVNC	RDPW	VNC	RDP	Citrix	TightVNC	RDPW	VNC
E-mail (attach pic)	457	457	554	2,623	550	439	460	1,962	343	439
E-mail (text)	518	518	554	4,113	513	485	604	2,601	722	572
Excel	698	698	1,345	6,839	729	706	1,345	5,332	703	1,145
Flash (30s)	22,404	22,404	6,903	1,801	13,387	11,453	449	976	6,062	921
ICQ	563	563	345	2,312	437	543	343	1,896	620	768
Java (stock chart)	4,912	4,912	5,762	9,022	7,095	1,358	5,799	7,135	1,688	-
Pictures	1,723	1,723	2,342	15,263	953	692	2,363	5,326	446	692
Power Point (slide show)	750	750	1,349	7,307	1,138	759	1,643	5,643	710	1,755
Power Point (no slide show)	699	699	442	4,410	-	649	468	3,560	-	2,125
Scmp	3,865	3,865	1,209	12,389	4,170	1,770	1,190	2,903	1,770	1,770
Sina_news	1,128	1,128	1,577	10,192	-	1,272	1,601	3,135	-	-
Sina_simplified	3,597	3,597	5,945	5,132	3,597	995	6,689	2,201	-	-
Telegraph	2,208	2,208	966	6,582	-	976	993	2,753	-	913
Word	572	572	1,007	6,315	509	587	1,021	4,192	414	1,008
Xinhua	4,743	4,743	2,061	5,299	6,066	1,159	1,342	2,098	3,095	1,159
Yahoo_traditional	2,461	2,461	7,002	18,866	-	1,377	6,911	4,983	-	-

Table 6 Data Packets in different environment (10Mbps & 56Kbps)

	Data Packets									
	Bandwidth: 10Mbps					Bandwidth: 56Kbps				
Test Case	RDP	Citrix	TightVNC	RDPW	VNC	RDP	Citrix	TightVNC	RDPW	VNC
E-mail (attach pic)	302	350	1,730	348	891	292	292	1,294	221	292
E-mail (text)	344	350	2,732	323	812	315	361	1,720	515	382
Excel	455	926	4,626	475	1,002	453	926	3,645	454	763
Flash (30s)	14,702	4,767	1,186	8,786	3,001	7,358	285	637	3,563	601
ICQ	388	213	1,685	290	501	379	244	1,364	426	501
Java (stock chart)	3,366	3,826	6,653	4,850	681	880	3,851	5,243	1,080	-
Pictures	1,173	1,615	10,465	622	1,173	450	1,620	3,623	257	450
Power Point (slide show)	496	928	4,926	752	1,045	482	1,077	3,866	442	1,292
Power Point (no slide show)	457	272	2,916	-	1,086	398	284	2,359	-	1,521
Scmp	2,663	860	8,645	2,762	2,663	1,214	840	1,995	1,214	1,214
Sina_news	767	1,261	7,175	-	-	863	1,254	2,156	-	-
Sina_simplified	2,557	4,122	3,512	2,557	-	595	4,756	1,523	-	-
Telegraph	1,493	678	4,462	-	2,576	633	673	1,923	-	676
Word	382	752	4,254	308	1,057	393	784	2,878	235	658
Xinhua	3,227	1,510	3,578	4,023	3,227	704	942	1,440	1,855	704
Yahoo_traditional	1,742	4,945	13,452	-	-	922	4,855	3,405	-	-

Table 7 Percentage of Data Packets in different environment (10Mbps & 56Kbps)

	% Data Packets									
	Bandwidth: 10Mbps					Bandwidth: 56Kbps				
Test Case	RDP	Citrix	TightVNC	RDPW	VNC	RDP	Citrix	TightVNC	RDPW	VNC
E-mail (attach pic)	66%	63%	66%	63%	64%	67%	63%	66%	64%	67%
E-mail (text)	66%	63%	66%	63%	67%	65%	60%	66%	71%	67%
Excel	65%	69%	68%	65%	64%	64%	69%	68%	65%	67%
Flash (30s)	66%	69%	66%	66%	65%	64%	63%	65%	59%	65%
ICQ	69%	62%	73%	66%	65%	70%	71%	72%	69%	65%
Java (stock chart)	69%	66%	74%	68%	70%	65%	66%	73%	64%	0%
Pictures	68%	69%	69%	65%	68%	65%	69%	68%	58%	65%
Power Point (slide show)	66%	69%	67%	66%	65%	64%	66%	69%	62%	74%
Power Point (no slide show)	65%	62%	66%	-	67%	61%	61%	66%	-	72%
Scmp	69%	71%	70%	66%	69%	69%	71%	69%	69%	69%
Sina_news	68%	80%	70%	-	-	68%	78%	69%	-	-
Sina_simplified	71%	69%	68%	71%	-	60%	71%	69%	-	-
Telegraph	68%	70%	68%	0%	55%	65%	68%	70%	-	74%
Word	67%	75%	67%	61%	66%	67%	77%	69%	57%	65%
Xinhua	68%	73%	68%	66%	68%	61%	70%	69%	60%	61%
Yahoo_traditional	71%	71%	71%	-	-	67%	70%	68%	-	-

Table 8 Average KBytes per packet in different environment (10Mbps & 56Kbps)

	Average KBytes/packet									
	Bandwidth: 10Mbps					Bandwidth: 56Kbps				
Test Case	RDP	Citrix	TightVNC	RDPW	VNC	RDP	Citrix	TightVNC	RDPW	VNC
E-mail (attach pic)	0.170	0.098	0.119	0.172	0.167	0.177	0.127	0.119	0.192	0.431
E-mail (text)	0.168	0.126	0.105	0.176	0.207	0.169	0.117	0.102	0.145	0.277
Excel	0.130	0.065	0.076	0.120	0.167	0.118	0.065	0.078	0.106	0.125
Flash (30s)	0.803	0.957	0.214	0.780	0.772	0.737	0.567	0.182	0.743	0.600
ICQ	0.175	0.078	0.076	0.150	0.177	0.115	0.089	0.080	0.131	0.143
Java (stock chart)	0.327	0.095	0.066	0.262	0.756	0.391	0.078	0.066	0.371	-
Pictures	0.203	0.066	0.170	0.342	0.151	0.204	0.065	0.132	0.245	0.529
Power Point (slide show)	0.438	0.108	0.081	0.364	0.181	0.463	0.072	0.085	0.391	0.105
Power Point (no slide show)	0.428	0.157	0.104	-	0.179	0.443	0.074	0.108	-	0.112
Scmp	0.519	0.077	0.194	0.421	0.852	0.342	0.083	0.170	0.342	0.304
Sina_news	0.088	0.068	0.122	-	-	0.101	0.064	0.147	-	-
Sina_simplified	0.461	0.788	0.277	0.461	-	0.497	0.787	0.227	-	-
Telegraph	0.534	0.065	0.217	-	0.752	0.251	0.062	0.193	-	0.620
Word	0.110	0.067	0.099	0.118	0.226	0.114	0.067	0.103	0.135	0.243
Xinhua	0.509	0.428	0.342	0.512	0.856	0.629	0.305	0.241	0.479	0.516
Yahoo_traditional	0.324	0.559	0.080	-	-	0.221	0.567	0.121	-	-

Table 9 Average Standard Deviation on Data Bytes transferred (10Mbps & 56Kbps)

		Average Standard Deviation										
	Test Case	RDP	Citrix	TightVNC	RDPW	VNC	RDP	Citrix	TightVNC	RDPW	VNC	
Bandwidth: 10Mbps	E-mail (attach pic)	3%	14%	3%	7%	9%	6%	2%	1%	14%	3%	Bandwidth: 56Kbps
	E-mail (text)	3%	1%	1%	6%	1%	4%	1%	1%	18%	12%	
	Excel	1%	2%	1%	3%	2%	1%	1%	1%	4%	50%	
	Flash (30s)	3%	1%	1%	6%	3%	1%	1%	1%	3%	3%	
	ICQ	21%	10%	9%	13%	16%	16%	31%	5%	22%	9%	
	Java (stock chart)	2%	18%	1%	17%	2%	1%	7%	4%	2%	2%	
	Pictures	103%	7%	1%	90%	4%	82%	4%	3%	50%	12%	
	Power Point (slide show)	10%	28%	1%	31%	2%	14%	1%	1%	51%	50%	
	Power Point (no slide show)	3%	29%	2%	-	2%	7%	5%	0%		47%	
	Scmp	5%	13%	2%	32%	1%	6%	16%	1%	6%	2%	
	Sina_news	24%	4%	1%	-	0%	37%	2%	4%		0%	
	Sina_simplified	12%	1%	1%	12%	0%	16%	8%	7%		0%	
	Telegraph	10%	3%	2%	-	6%	40%	1%	8%		7%	
	Word	4%	4%	2%	14%	1%	7%	4%	4%	9%	1%	
	Xinhua	17%	2%	3%	23%	1%	2%	24%	3%	32%	14%	
	Yahoo_traditional	8%	2%	1%	-	0%	37%	1%	4%		0%	

Table 10 Time for completion in different environment (100Mbps & 56Kbps)

		Time for completion										
	Test Case	RDP	Citrix	TightVNC	RDPW	VNC	RDP	Citrix	TightVNC	RDPW	VNC	
Bandwidth: 10Mbps	E-mail (attach pic)	47.7	50.7	50.2	69.3	68.0	45.8	50.2	50.5	33.0	70.0	Bandwidth: 56Kbps
	E-mail (text)	49.7	59.8	51.0	69.2	70.0	49.4	60.5	51.5	74.6	68.1	
	Excel	61.7	65.1	67.8	69.3	69.9	63.4	65.1	75.2	67.1	69.4	
	Flash (30s)	30.0	30.4	30.1	36.2	67.8	960.3	30.1	149.2	533.6	67.8	
	ICQ	47.3	46.2	47.3	31.7	69.1	45.9	45.3	50.5	37.7	69.1	
	Java (stock chart)	63.3	70.5	66.5	63.1	69.1	63.6	70.2	69.8	74.6	69.1	
	Pictures	57.2	60.1	65.2	68.3	57.2	77.4	61.5	68.5	54.0	71.0	
	Power Point (slide show)	61.8	61.0	62.3	70.2	63.5	62.4	61.2	78.4	78.4	68.4	
	Power Point (no slide show)	60.2	63.9	61.3	-	70.1	59.3	70.2	64.4	-	68.4	
	Scmp	64.0	62.9	70.2	69.4	64.0	73.2	63.0	142.2	73.2	73.2	
	Sina_news	62.9	69.5	70.6	-	-	65.6	69.9	80.4	-	-	
	Sina_simplified	66.4	71.3	69.9	66.4	-	81.4	70.0	130.2	-	-	
	Telegraph	63.6	64.6	63.6	-	65.9	72.8	63.0	72.4	-	65.9	
	Word	68.1	69.4	66.1	68.9	70.1	68.9	70.1	78.3	67.0	68.9	
	Xinhua	65.1	63.0	65.0	69.2	65.1	80.0	63.0	156.1	248.6	80.0	
	Yahoo_traditional	64.5	64.4	70.6	-	-	68.5	64.5	80.2	-	-	

Table 11 Data on 10Mbps environment in narrow downed tests

		Average (Kb)					**Time (seconds)**				
	Test Case	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**
Bandwidth: 8kbps	E-mail (text)	50.3	77.1	249.5	77.1	249.5	65.0	67.5	63.8	63.0	69.6
	Flash (30s)	10,812.4	4,253.8	4,945.5	4,253.8	4,945.5	76.3	30.6	38.4	32.8	69.7
	ICQ	42.6	370.9	135.8	370.9	135.8	18.2	18.2	18.5	19.6	68.6
	Pictures	22.3	66.2	260.2	66.2	260.2	25.4	30.2	23.0	21.9	66.7
	Xinhua	44.0	199.4	4,058.4	199.4	4,058.4	26.0	30.3	23.8	22.2	68.5
		Total Packets					**Data Packets**				
	Test Case	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**
	E-mail (text)	466	622	3,468	399	1,207	259	367	2,304	265	812
	Flash (30s)	14,502	6,820	3,551	6,016	1,609	9,410	4,451	2,480	3,843	1,086
	ICQ	364	364	576	650	768	266	266	397	449	501
	Pictures	154	168	1,382	225	1,723	96	99	916	152	1,173
	Xinhua	218	159	1,001	632	4,743	137	94	678	412	3,227
		% Data Packet					**Average KBytes/packet**				
	Test Case	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**
	E-mail (text)	56%	59%	66%	66%	67%	0.108	0.068	0.095	0.193	0.207
	Flash (30s)	65%	65%	70%	64%	67%	0.746	0.823	0.251	0.707	3.074
	ICQ	73%	73%	69%	69%	65%	0.117	0.117	0.078	0.571	0.177
	Pictures	62%	59%	66%	68%	68%	0.145	0.143	0.204	0.294	0.151
	Xinhua	63%	59%	68%	65%	68%	0.202	0.133	0.326	0.316	0.856

Table 12 Data on 8Kbps environment in narrow downed tests

		Average (Kb)					**Time (seconds)**				
	Test Case	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**
Bandwidth: 8kbps	E-mail (text)	43.9	37.0	76.8	-	52.1	65.8	62.2	66.8	-	69.2
	Flash (30s)	-	176.3	119.4	-	109.0	-	82.9	182.1	-	100.6
	ICQ	52.4	14.2	27.8	-	-	59.6	18.0	26.4	-	-
	Pictures	191.4	18.5	35.6	64.0	147.7	127.7	25.8	38.7	36.1	140.1
	Xinhua	185.1	18.5	32.3	164.0	87.4	95.9	32.7	48.8	136.2	68.2
		Total Packets					**Data Packets**				
	Test Case	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**
	E-mail (text)	520	420	803	-	425	294	234	537	-	275
	Flash (30s)	-	450	572	-	457	-	241	365	-	302
	ICQ	906	183	359	-	-	579	129	243	-	-
	Pictures	1,773	122	255	322	460	1,142	77	167	195	297
	Xinhua	532	145	167	519	423	334	74	107	303	300
		% Data Packet					**Average KBytes/packet**				
	Test Case	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**	**RDP**	**Citrix**	**TightVNC**	**RDPW**	**VNC**
	E-mail (text)	57%	56%	67%	-	65%	0.084	0.088	0.096	-	0.123
	Flash (30s)	-	54%	64%	-	66%	-	0.392	0.209	-	0.239
	ICQ	64%	70%	68%	-	-	0.058	0.078	0.077	-	-
	Pictures	520	420	803	-	425	294	234	537	-	275
	Xinhua	-	450	572	-	457	-	241	365	-	302

4.2. References

[1] Remote Desktop Protocol (RDP) Features and Performance – White Paper. Microsoft Windows 2000 Server.

[2] ITU-T Recommendation T.120

[3] GDI + http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdicpp/GDIPlus/GDIPlusreference/flatapi.asp

[4] New Font Format http://msdn.microsoft.com/library/default.asp?url=/library/en-us/w98ddk/hh/w98ddk/display_1fav.asp

[5] Remote Desktop Protocol http://msdn.microsoft.com/library/default.asp?url=/library/en-us/termserv/termserv/remote_desktop_protocol.asp

[6] What makes VNC different from other systems? http://www.realvnc.com/why.html

[7] VNC – How it works http://www.realvnc.com/howitworks.html

[8] Introduction to TightVNC http://www.tightvnc.com/intro.html

[9] TightVNC Change Logs http://www.tightvnc.com/changelogs.html

[10] VNCViewer Manual Page http://www.tightvnc.com/vncviewer.1.html

[11] Citirx History http://www.citrix.com/