Manual of AwareWare Demonstration System

1. Introduction

1.1. AwareWare

AwareWare is the adaptation middleware we proposed to support external adaptation functions for component-based applications. External adaptation mechanisms are responsible for monitoring conditions that trigger adaptation, decide when, how and where to adapt application behaviors, and execute the adaptation plan on a running system. It is believed that adaptation helps achieve robust middleware services, which are especially important for enhancing scientific productivity and for facilitating research and education collaborations through the sharing of data, instruments, and computing resources. Therefore, this Phase I demonstration focuses on showing the performance advantages of using AwareWare to support distance education across heterogeneous environments.

AwareWare is a middleware situated between Operating System (OS) and adaptive applications. Its architecture consist of the following modules as depicted in Fig. 1: awareness acquisition or measurement tools, awareness manager, consistency manager, adaptation decision module, and reconfiguration interface.

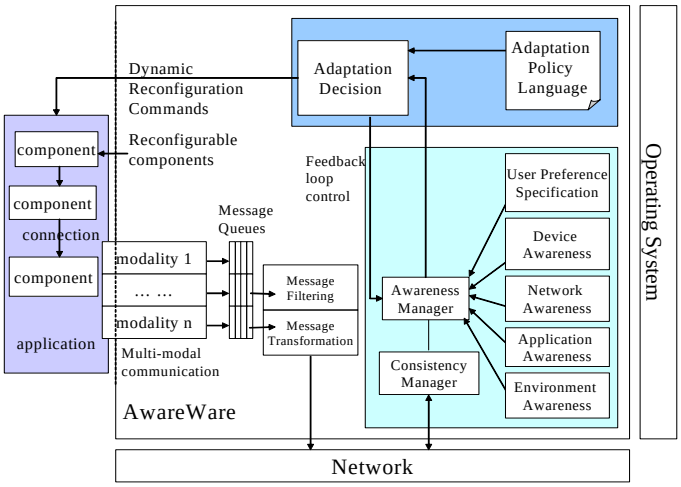


Fig. 1 Proposed middleware architecture of *AwareWare*

1.2. Demonstration System

The demonstration illustrates the major functionality of AwareWare. It is established based on the architecture of the proposed AwareWare as depicted in Fig. 2. It contains most of the major modules we introduced in Section 1.1. The awareness acquisition or measurement tools measure information about networks, devices, end-user preferences, application internal states, and physical environments. The aware manager organizes these tools and provides middleware interfaces for adaptive decisions. The adaptation decision module takes awareness data as inputs and initiates dynamic reconfiguration and parameter tuning commands to the applications. It also provides a feedback loop to the awareness manager, which in turn controls the behavior of the measurement tools.

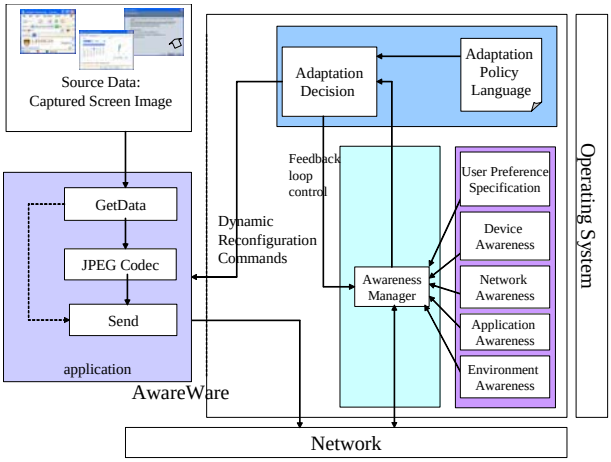


Fig. 2 Architecture of the Demonstration System

We use remote application sharing for distance education as an example application in the demonstration. Remote application sharing (a.k.a. remote screen sharing) is a distributed system that captures screen images of a desktop or an application and sends them to remote computers for display. There are multiple commercialized products, which include Microsoft NetMeeting [xxx], WebEx [xxx], etc. However, they have limited adaptation behaviors, e.g., they only allow an end-user to specify the network connection type for captured screen images. In this demonstration, we try to integrate the functionality of AwareWare and a remote application sharing to show the adaptation capability of the proposed AwareWare in a network with variations on available bandwidth.

In the implemented prototype of AwareWare, awareness measurement tools measures the resource information of the available bandwidth for network links and the end user preferences. The awareness measurement tools can be provided with applications or come with the AwareWare as a standalone module. In this demonstration, two different types of Awareness information are referred in the decision-making process. From the awareness information measured by a measurement tool for end-user preference, the AwareWare obtains the minimum picture quality a user can accept. Meanwhile, the application replies the frame rate of the running interface sharing system. With these two types of awareness information available, the decision engine makes decision on the proper picture quality. The decision is transmitted to the application to control the compression component. As a result, the quality of the picture will follow the user”s preference whenever it is available. When the user”s preference is satisfied or no user preference is available, the decision engine tries it best to satisfy a hardwired minimum frame rate. Therefore, user”s preference has higher priority than the minimum frame rate requirement in the decision-making policy.

To transfer the measured awareness information to the decision engine, the measurement tools will transfer the awareness information to the awareness manager, through which, different applications can share the awareness information. In the demonstration system, we use SNMP server as the awareness manager to simplify the implementation.

According to the requirements of applications, awareness manager drives measurement tools to measure awareness information with either on-demand polling or periodical reporting method. We call the on-demand polling method PULL method and the periodical reporting method PUSH method. No matter which method is applied, the awareness manager tries to provide necessary awareness information to the adaptation decision module.

By obtaining the awareness information from the awareness manager, the adaptation decision module can help applications to make decisions in connecting proper components and configuring certain parameters at run time. In the demonstration system, the adaptation decision module can decide whether to use a JPEG codec to compress captured screen images before transmission. It can also determine proper quality parameter to control the compression ratio of shared images according to the actual frame rate and user preferences. We illustrate the relationship between the parameter that controls compression quality and the consumed bandwidth in Table 1. According to Table 1, the adaptation decision module determines the optimal quality of image compression according to the user preference and the frame rate requirement.

Table 1. Relations between the quality parameter and the bandwidth consumption of the JPEG codec

Quality Parameter	Bandwidth Consumption (bps) when the sampling frequency is 10 frames/second
1	270k
5	400k
10	630k
15	760k
20	850k
25	980k
50	1000k
80	2000k
100	3000k

1.3. Demonstration Network Configuration

In our test for the demonstration system, we use softperfect bandwidth manager to control the available bandwidth between two computers that are connected to high-speed internet to show the adaptation capability provided by AwareWare. The two computers are connected into the 1GHz campus network of Lehigh University. We install the softperfect bandwidth management tool to control the available bandwidth between the server and the client by using the following two rules.

```
“Rule name”=rule1
“Direction”=both
“Rate Limit”=15.2KB/s
“Protocol”=TCP/UDP
```

```
“Source”=localhost:any
“Destination”=all:any
“Interface”=any
```

```
“Rule name”=rule2
“Direction”=both
“Rate Limit”=15.2KB/s
“Protocol”=TCP/UDP
“Source”=all:any
“Destination”=localhost:any
“Interface”=any
```

2. Installation Manual

2.1. Install Middleware System

2.1.1. Install the SNMP Server

In this section, we explain how to install the SNMP server in Windows XP and configure it properly. In this demonstration, we run the SNMP server on the interface sharing server computer, which will act as SNMP client and server simultaneously. The interface sharing client computer will act as an SNMP client to register certain information into the SNMP server running at the server computer. To achieve this goal, we will use IP address of the server computer (128.180.123.17) to configure the SNMP service.

By default, Windows XP does not have SNMP installed. Therefore, install the SNMP services first. For that, go to "Control panel", "Add\Remove Programs". Click "Add/Remove Windows components". Check the checkbox "Management and Monitoring Tools". Then on Detail, check "Simple Network Management Protocol". Click “Next” and complete the installation. Most probably, you will need Windows XP CD inserted to complete the installation.

After the installation, go to check the services and make sure you have two new services "SNMP Service" and "SNMP Trap Service". "SNMP Service" corresponds to the service file "SNMP.EXE" and "SNMP Trap Service" corresponds to the service file "SNMPTRAP.EXE".

To configure, on Services (Management Console Tool), right click on "SNMP Service" and click "Properties". Properties dialog will open up as depicted in Fig. 3.

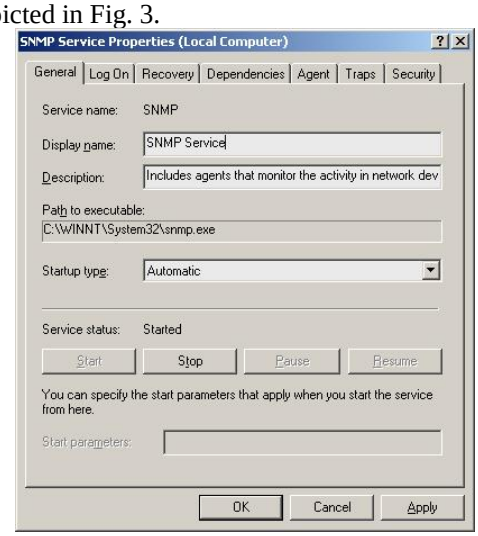


Fig. 3 Properties of SNMP service

Select the tab "Traps". Type "public" on "community name" combo box and click "Add to list" button. Then to fire traps to our same computer, click "Add" and add the IP address "127.0.0.1". You also need to enable the client computer to access the SNMP server by adding its IP address (128.180.123.75) in the dialog by similar means.

After this, go to "Log On" tab and enable the "Profile 1" by clicking "Enable" button. Also, make sure "Local system account" radio button appears selected.

Next, we can configure security of the SNMP server in the tab "Security". The security tab will look like Fig. 4. Check the checkbox "Send authentication trap". Click "Add" under "Accepted community names". On appearing dialog box, select "READ WRITE" on "Community rights" and enter "public" on "Community name". If you want to accept requests only from a particular client, then select "Accept SNMP packets from these hosts" and click "Add" underneath that. Type the host name or IP address and click "OK" on the appearing dialog box. If you select accept SNMP packets from these hosts, please make sure that the IP addresses of both the server computer and the client

computer are added. After all this, click "Apply" or "OK" and close the "SNMP Service" Properties dialog. If the service is not started, right click and start it.

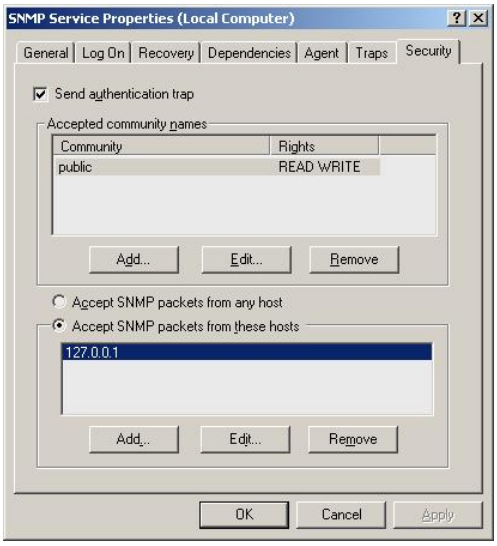


Fig. 4 Properties of SNMP service

After finishing "SNMP Service" configuration, select "SNMP Trap Service" and open up the Properties dialog by right clicking on it. On the "Log On" tab, configure as for the service before. Then "Apply" or "OK", and close the "SNMP Trap Service" Properties dialog.

After this, the SNMP service should be ready to work in our demonstration.

2.1.2. Server side program--ShowMeLive

At the server computer, download and run the setup file “AwareWareDemo1_server_setup.exe”. The server program and AwareWare library will be automatically installed on your computer. You need to double click the “AwareWare_SNMP.reg” in the folder of installation (“C:\Program Files\AwareWare Demonstration Server\Server” is the default value). If you changed the default installation folder for the application, please make sure that you change the last line in the “AwareWare_SNMP.reg” before double clicking, such that the SNMP agent can be found. Following is the content of the “AwareWare_SNMP.reg”.

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents]
"1"="SOFTWARE\\Microsoft\\LANManagerMIB2Agent\\CurrentVersion"
"2"="SOFTWARE\\Microsoft\\RFC1156Agent\\CurrentVersion"
"3"="SOFTWARE\\Microsoft\\HostMIB\\CurrentVersion"
"4"="SOFTWARE\\Microsoft\\SNMPMIB\\CurrentVersion"
"5"="SOFTWARE\\Microsoft\\SNMP_EVENTS\\CurrentVersion"
"6"="SOFTWARE\\Microsoft\\ACS\\CurrentVersion"
"7"="SOFTWARE\\Microsoft\\IGMPMibAgent\\CurrentVersion"
"8"="SOFTWARE\\Microsoft\\IPMulticastMibAgent\\CurrentVersion"
"9"="SOFTWARE\\Microsoft\\IPXMibAgent\\CurrentVersion"
"10"="SOFTWARE\\Microsoft\\IASAgent\\CurrentVersion"
"11"="SOFTWARE\\SNMP\\CurrentVersion"

[HKEY_LOCAL_MACHINE\SOFTWARE\SNMP\CurrentVersion]
"Pathname"="C:\\Program Files\\AwareWare Demonstration Server\\server\\SNMPAgent.dll"
```

Next, please edit the file “admin.ini”, “conf.ini” and “developer.ini” to make sure IP address of SNMP server is correctly configured. Assume your IP address of your server computer is 128.180.123.17. You should configure the “admin.ini” as follows.

```
[System]
SNMPServer  = 128.180.123.17
[Awareness]
len = 3
[Bandwidth]
IPAddress   = 128.180.123.17
SNMPNode    = .1.3.6.1.4.1.15.0.4.1
MIBEntryNO  = 5
Method    =  push
minimun    =   0
maximum    =   500
```

```
[CPUUsage]
IPAddress   = 128.180.123.17
SNMPNode    = .1.3.6.1.4.1.15.0.2.1.2
MIBEntryNO  = 2
Method    =  pull
frequency   =   200
[UserPref]
IPAddress   = 128.180.123.17
SNMPNode    = .1.3.6.1.4.1.15.0.5.1
MIBEntryNO  = 7
Method    =  pull
frequency   =   200
```

In this file, we configured the SNMP service is provided by computer with IP address 128.180.123.17. This application will use three pieces of awareness information, Bandwidth, CPU usage, and user preference respectively. Their SNMP node address, MIB node entry number, updating method and minimum updating frequency is configured in the "admin.ini".

The "conf.ini" helps the decision engine to find the registered user preference information in the MIB information base. It looks like this:

```
[Configure]
RemoteHost = 128.180.123.17
PreferOid  = .1.3.6.1.4.1.15.0.5.1
```

The "developer.ini" should looks like:

```
[Variable]
len = 2
v1  = UserPref
remoteHost = 128.180.123.17
v2  = CPUUsage
[Decision]
number = 2
d1 = JPEGQuality
d2 = CPUInf
JPEGQuality = (UserPref+100/10-2*5)*10/(7+3)
CPUInf = CPUUsage * (2500 + 25) / 100
```

After revising these two files, the server has been configured correctly.

2.1.3. Client side program--TeachLive

At the client computer (the computer you would like to run the client of the interface sharing application), Download and run the setup file "AwareWareDemo1_client_setup.exe". The server program and AwareWare library will be automatically installed on your computer.

Config the "conf.ini" as follows such that the client can discover and use the SNMP server located at the server computer.

```
[Configure]
RemoteHost = 128.180.123.17
PreferOid  = .1.3.6.1.4.1.15.0.5.1
```

2.1.4. Install the softperfect bandwidth management tool

Download the trial version of the software from "http://www.softperfect.com/download/bwmsetup.exe". Execute the program to install the software on the interface sharing server computer. Follow the instruction to restart the computer before run the installed bandwidth management tool. Log into the bandwidth management tool with user name "local host" and password "" as shown in Fig. 5.

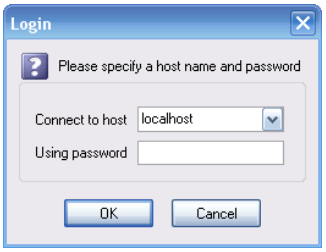


Fig. 5 Login window of the softperfect bandwidth management tool

Add a rule into the system to control the bandwidth between the server computer and the client computer to a certain value and enable the rule as shown in Fig. 6.

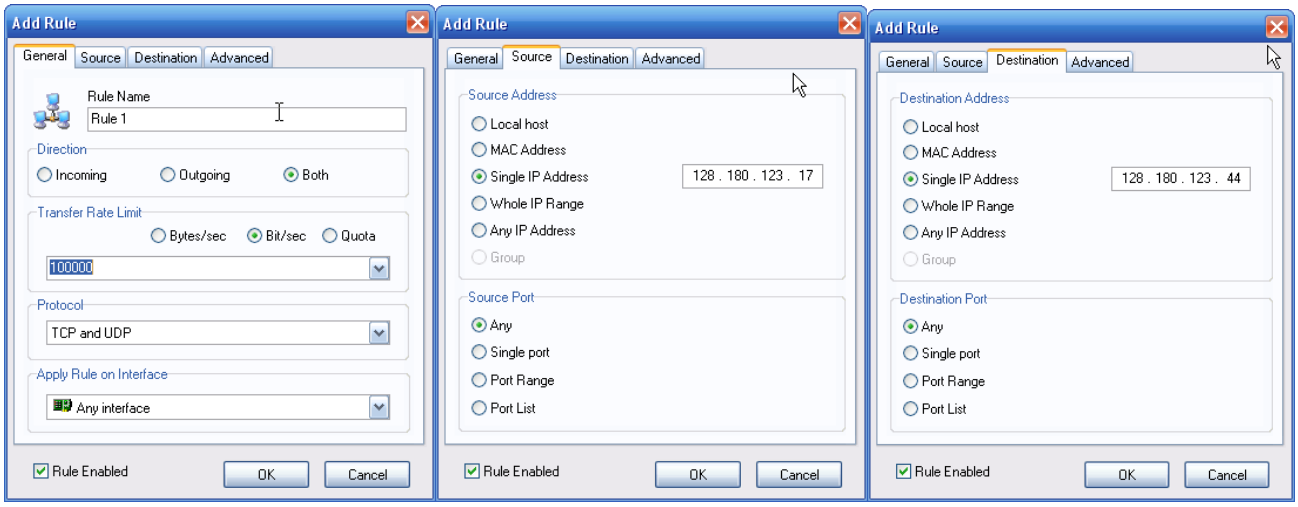


Fig. 6 Adding rules into the softperfect bandwidth management tool

Now you can control the available bandwidth between the two testing computers under the bandwidth management software.

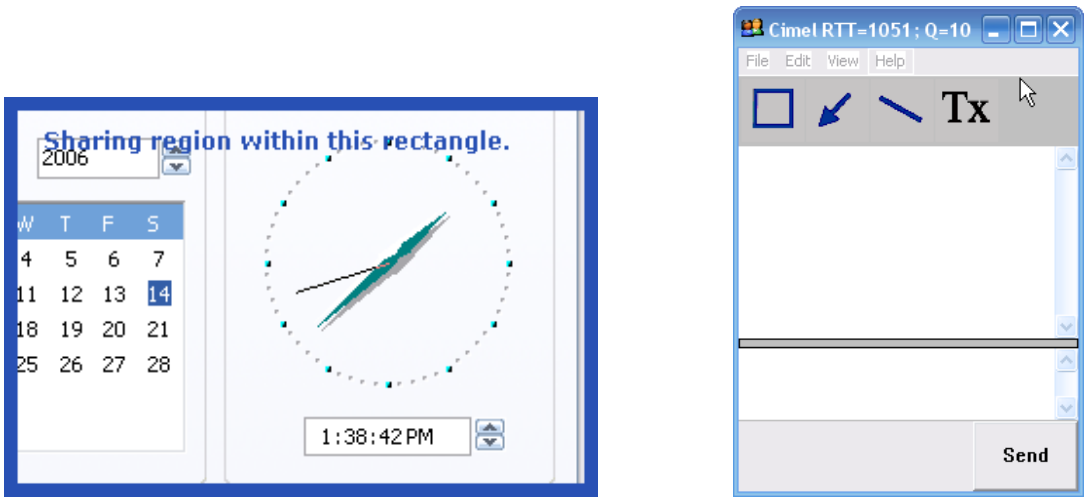
3. Execution Manual

3.1. Connect the Remote and Local Computer

In order to run the demonstration, two PCs are necessary. One of the PC is the server machine, at which the local experiment is conducted. The server machine will execute the ShowMeLive program to share its view of the experiment with other clients online. The other PC serves as a client, where the TeachLive program is running. The first step of the demo is to connect the server machine and the client machine. In order to demonstrate the bandwidth adaptation capability of our middleware, we use softperfect bandwidth management tool to control the available bandwidth between the server and the client. We also obtain the IP address for both computers during the connection process. In the rest of this manual, we refer to these two IP address as 128.180.123.17 and 128.180.123.75 respectively.

3.2. Run Interface Sharing Server at the Server Computer

After queries are responded by the sensor network, we can run the interface sharing server program ShowMeLive at the server computer. It will automatically load the middleware components. The interface is shown in Fig. 12. Put the application you want the remote client to see in the blue box as shown in Fig. 12(a). In this manual, we recommend using the windows clock, through which the remote user can easily identify the changes for both image quality and frame rate due to bandwidth variations. In this demonstration, the blue box is set to be 300-pixel in width by 200-pixel in height.



(a) (b)

Fig. 12 Interface sharing server interface

Click the “wait for connection” in the file menu of the ShowMeLive window. Input the IP address of the client

computer 128.180.123.17 as shown in Fig.13.

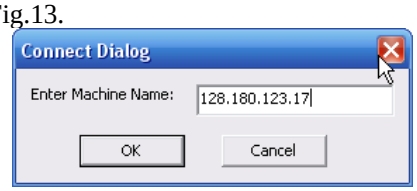


Fig. 13 Waiting dialog for ShowMeLive Program

Now the server is waiting for the client to connect.

3.3. Run Interface Sharing Client at the Client Computer

The last step of this demo is to run the interface sharing client program at the client computer and connect to the interface-sharing server to share the TinyDB experiment result. Execute TeachLive program at the client computer. The interface shown in Fig. 14 will appear.

Fig. 14 TeachLive interface

Click the menu item “connect” in file menu. Input the IP address of the server computer 128.180.123.75 in the dialog box as shown in Fig. 15.

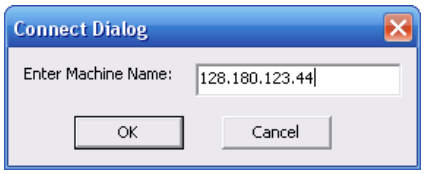


Fig. 15 Connect dialog of TeachLive

An image of the area that is bounded by the blue box should appear as shown in Fig. 16.

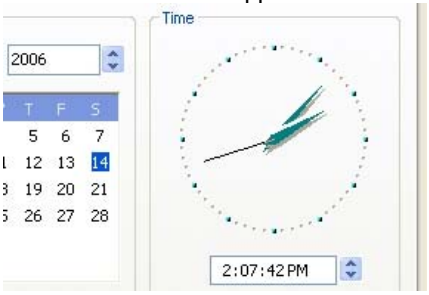


Fig. 16 Viewable shared images at the client side

At the server side, the average frame rate and can be viewed in the caption bar as shown in Fig. 17.

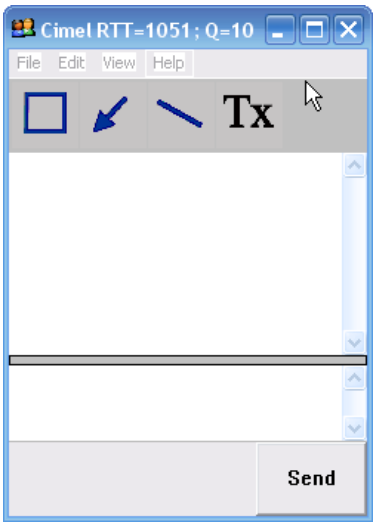


Fig. 17 Average RTT and image quality parameter are shown in the title bar of the server application

3.4. Change the user preference in the client interface to affect the quality of the shared images

According to the available bandwidth, the frequency of the viewed interface varies. Adjusting the available bandwidth by changing rules at the server side, at the client computer, users will observe sampling quality changes for the shared view. It demonstrated that the AwareWare system could be adaptive to the available bandwidth variations.

Now at the client side, user should have seen the shared interface images from the server side. The server program begins by sending the highest quality images. Due to the frame rate requirement, in a network with limited bandwidth, the image quality may begin to drop. When the user feels that the quality is not acceptable, he press one or several "+"s and click "Send" button at the client application to set a user preference in the SNMP server. The client computer will read current quality parameter from the SNMP server and set user preferred image quality parameter in the SNMP server as the summation of the current image quality parameter and the number of "+"s that the user just pressed. For example, if the current image quality control parameter obtained from the SNMP server is 10 and user pressed 10 "+"s before press the send button, the user preferred image quality will be set as 20 in the SNMP server. By getting this user preferred quality parameter from the SNMP server, the decision engine knows the lowest quality that the remote user can accept. Based on this user preference, the decision engine will decide the proper parameter that controls the image compression quality according to the bandwidth variation. If the bandwidth is not enough to support communication of shared images with the minimum frame rate and the lowest acceptable quality, the decision engine will override the minimum frame rate by setting the quality control parameter as the user preferred quality parameter. However, when the bandwidth is enough, both the minimum frame rate and the lowest user preferred quality will be satisfied. When the user observe a smooth image sequence of the server computer, they can try to press multiple "+"s and the "send" button to increase the image quality. This information will be recorded in the SNMP server and affect the performance of the interface sharing application.

3.5. Adjust the available bandwidth in the softperfect bandwidth management tool to simulate a network with changing bandwidth

First, in the softperfect bandwidth management tool, configure the bandwidth from the server to the client to 600kbps. After the configuration, the softperfect bandwidth management tool looks like Fig. 18.

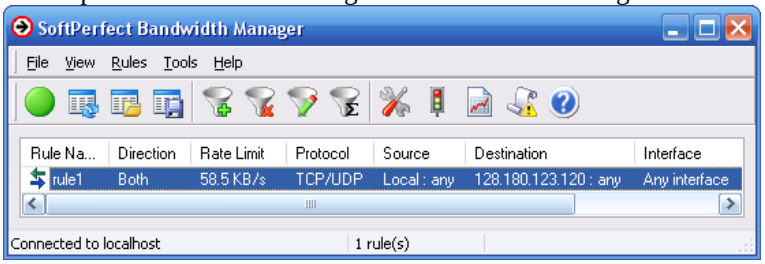


Fig. 18 The softperfect bandwidth management tool interface after configuring the bandwidth between server and client as 600kbps

From the windows resource monitor, it can be observed that the bandwidth between the server and the client are

controlled very well at 600kbps in Fig. 19.

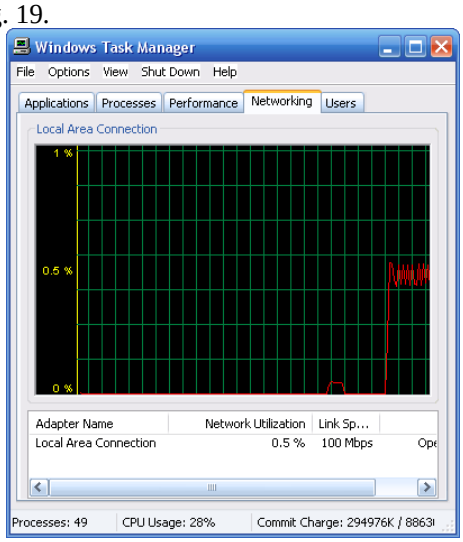
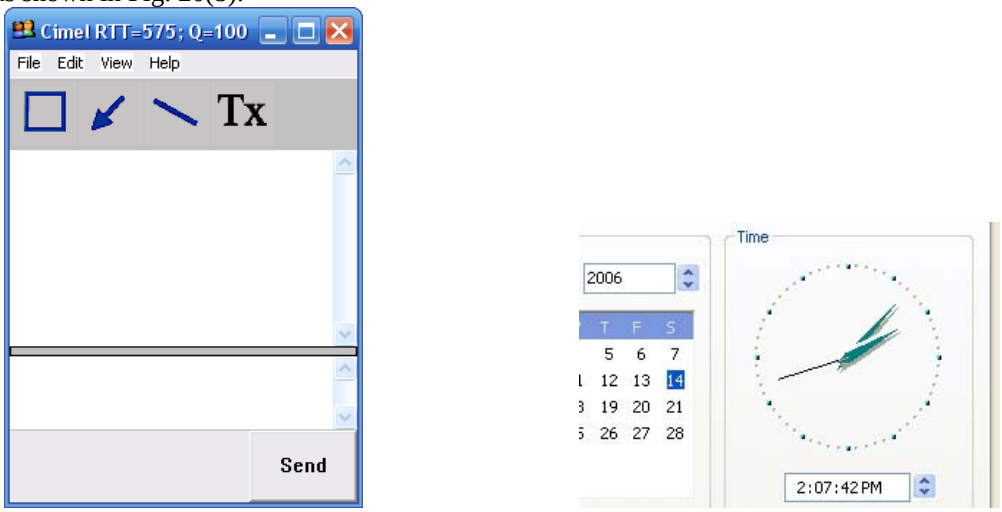


Fig. 19 Windows resource monitoring when the bandwidth between the server and client is controlled at 600kbps.

Since the size of shared image is quite small, 600kbps is enough for the interface sharing application to satisfy the minimum frame rate (2f/s) with least image compression (image quality control parameter equals to 100). The server and client side view is shown in Fig. 20. From Fig. 20(a), it is clear that the minimum frame rate is almost satisfied with image compression quality equals to 100. Therefore, the client can see a smooth clock at the client computer with high quality as shown in Fig. 20(b).



(a) (b)

Fig. 20 Server and client view of the remote interface sharing application when bandwidth equals to 600kbps

Next, in the softperfect bandwidth management tool, configure the bandwidth from the server to the client to 60kbps. After the configuration, the softperfect bandwidth management tool looks like Fig. 21.

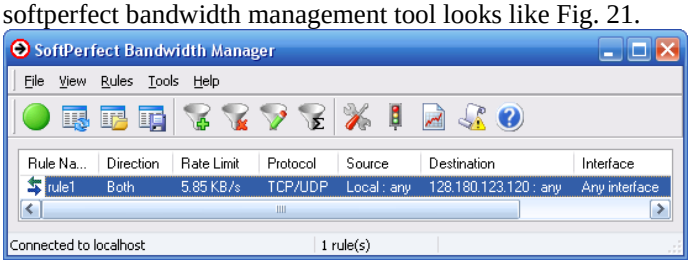


Fig. 21 The softperfect bandwidth management tool interface after configuring the bandwidth between server and client as 60kbps

From the windows resource monitor, it can be observed that the bandwidth between the server and the client are controlled very well at 60kbps in Fig. 22.

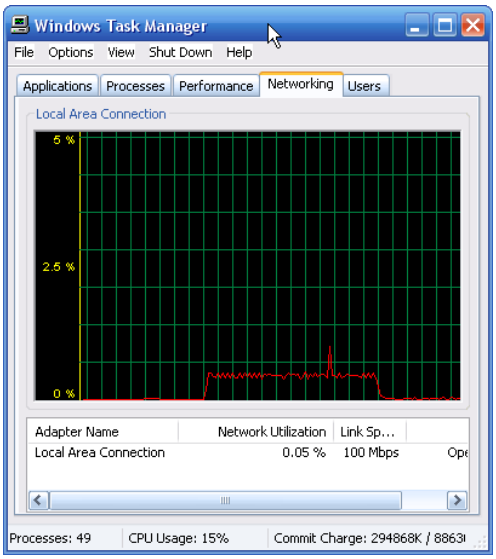
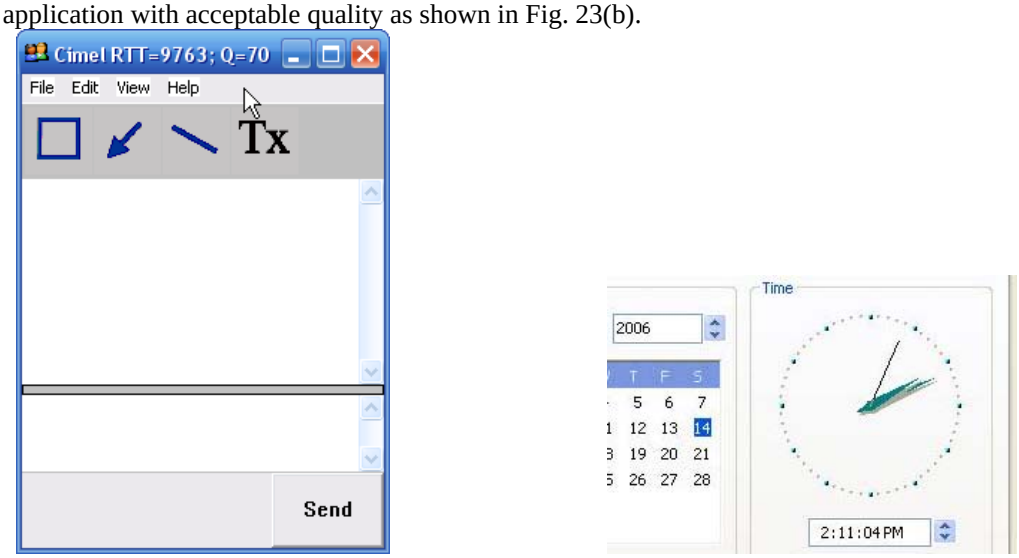


Fig. 22 Windows resource monitoring when the bandwidth between the server and client is controlled at 60kbps.

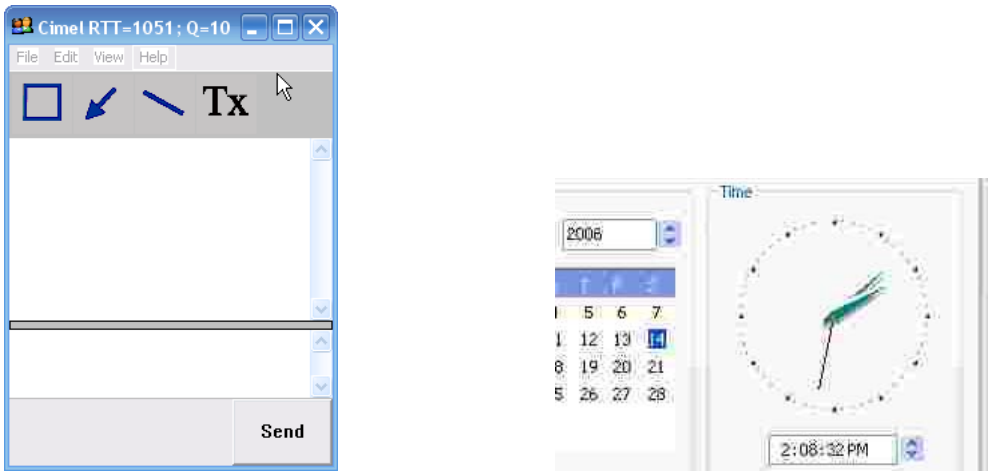
Since the bandwidth is too small to satisfy both minimum frame rate requirement (2f/s) while using the best quality image compression, the decision engine will change its decision on the proper quality of image compression according to the policy we described earlier. The server and client side view at the time when the bandwidth has just been adjusted is shown in Fig. 23. From Fig. 23(a), it is obvious that the decision engine has already adjusted the image quality by reducing the image compression quality from 100 to 70. However the average frame rate is only 0.1f/s with image compression quality equals to 70. Therefore, the client can see only bursty (0.1f/s) update if the clock in the application with acceptable quality as shown in Fig. 23(b).



(a) (b)

Fig. 23 Server and client view of the remote interface sharing application when bandwidth equals to 60kbps

Since the minimum frame rate is not satisfied and there is still no user preference available, the decision engine keeps reducing the compressed image quality to increase the frame rate in the system. The server and client side view at the time when the bandwidth has just been adjusted is shown in Fig. 24. From Fig. 24(a), it can be seen that the image quality is reduced from 70 to 10 while the frame rate of the communication is increased to almost 1f/s. However, it can be seen in Fig. 24(b) that the image quality is very poor now.



(a) (b)

Fig. 24 Server and client view of the remote interface sharing application when bandwidth equals to 60kbps

At this time, the client feels unhappy about the shared image quality. Therefore, the client types in 10 "+"s in the lower part of the client application and clicks "send" button to transfer his preference as shown in Fig. 25.

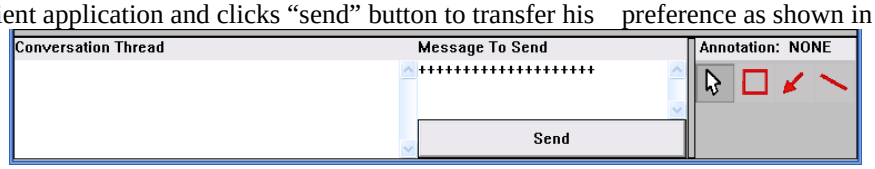
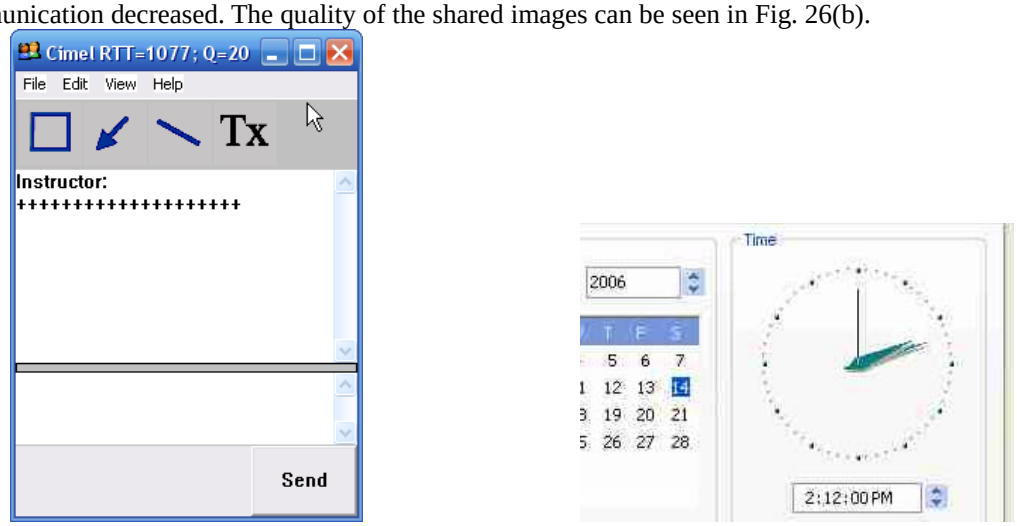


Fig. 25 User preference seting

The user preference is registered in the SNMP server. As a result, the decision engine will satisfy this requirement with highest priority. The server and client side view at the time when the bandwidth has just been adjusted is shown in Fig. 26. From Fig. 26(a), it can be seen that the image quality is increased from 10 to 20. As a result, the frame rate of the communication decreased. The quality of the shared images can be seen in Fig. 26(b).



(a) (b)

Fig. 26 Server and client view of the remote interface sharing application when bandwidth equals to 60kbps

Finally, in the softperfect bandwidth management tool, configure the bandwidth from the server to the client to 300kbps. After this configuration, the softperfect bandwidth management tool looks like Fig. 27.

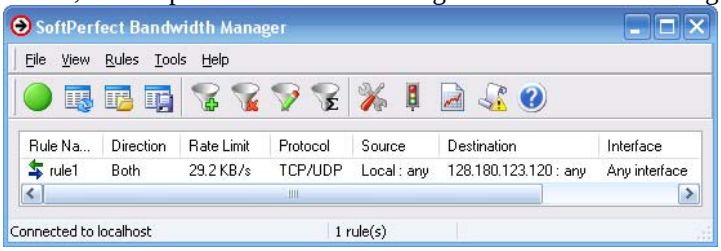


Fig. 27 The softperfect bandwidth management tool interface after configuring the bandwidth between server and client as 300kbps

From the windows resource monitor, it can be observed that the bandwidth between the server and the client are controlled very well at 600kbps in Fig. 28.

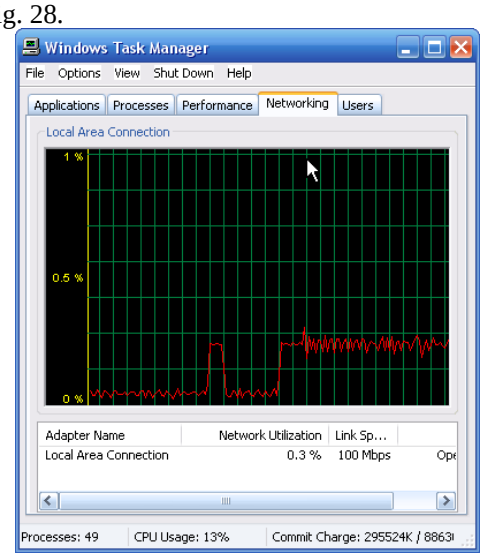
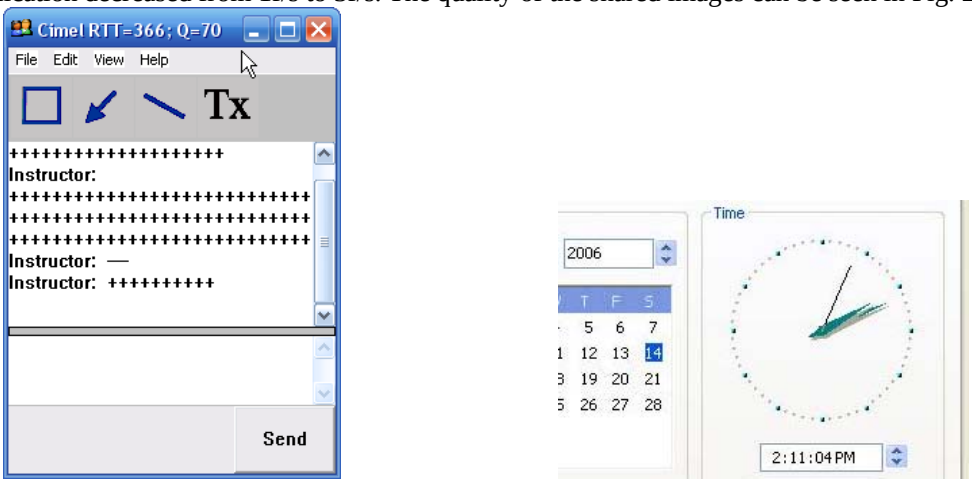


Fig. 28 Windows resource monitoring when the bandwidth between the server and client is controlled at 300kbps.

The user preference is registered in the SNMP server. As a result, the decision engine will satisfy this requirement with highest priority. The server and client side view at the time when the bandwidth has just been adjusted is shown in Fig. 29. From Fig. 29(a), it can be seen that the image quality is increased from 20 to 70. As a result, the frame rate of the communication decreased from 1f/s to 3f/s. The quality of the shared images can be seen in Fig. 29(b).



(a) (b)

Fig. 26 Server and client view of the remote interface sharing application when bandwidth equals to 60kbps