

ACQUA

Desktop Prototype

User Guide



<https://project.inria.fr/acqua/>

v0.0

Project Overview

ACQUA is an Application for prediCting Quality of User Experience at Internet Access. It was supported by the French [ANR CMON](#) project on collaborative monitoring and is currently supported by both the Inria ADT ACQUA and the ANR Project [BottleNet](#).

ACQUA presents a new way for the evaluation of the performance of Internet access. Starting from network-level measurements as the ones we often do today (bandwidth, delay, loss rates, etc), ACQUA targets the estimated quality of experience related to the different applications of interest to the user without the need to run them (e.g. estimated Skype quality, estimated video streaming quality). An application in ACQUA is a function that links the network-level measurements to the expected quality of experience. Statistical machine learning techniques are used to establish such link between measurements both at the network level and device level, and estimations of the Quality of Experience for different Internet applications. The required data for such learning can be obtained either by controlled experiments as we did in a recent communication on Skype Quality of Experience, or by soliciting the crowd (i.e., crowdsourcing) for combinations of measurements and corresponding experience. In its first version (the version available online), ACQUA was concentrating on delay measurements at the access and on the detection and estimation of the impact of delay anomalies (local problems, remote problems, etc). The current work is concentrating on using the ACQUA principle in the estimation and prediction of the quality of experience of main user's applications (we refer to [this video](#) and [this paper](#) for more details).

ACQUA Desktop Prototype

User guide

This simple prototype performs active measurements to establish network conditions (in terms of Bandwidth, Loss Rate and RTT) and computes QoE outputs for Internet applications (Skype and Video-like output).

It is based on a distributed architecture (Client / Server) where both hosts performs symmetrical measurements.

The Client is a C# GUI based application while the Server is a Java daemon.

Application Setup

The project files are divided in two subsets:

- **main application files**, where the ACQUA measurement platform source code is defined;
- **demo platform files**, the platform to support standalone testing simulating network performances via DummyNet (network emulator). Contains a dynamic web page and some CGI scripts to set the underlying network conditions.

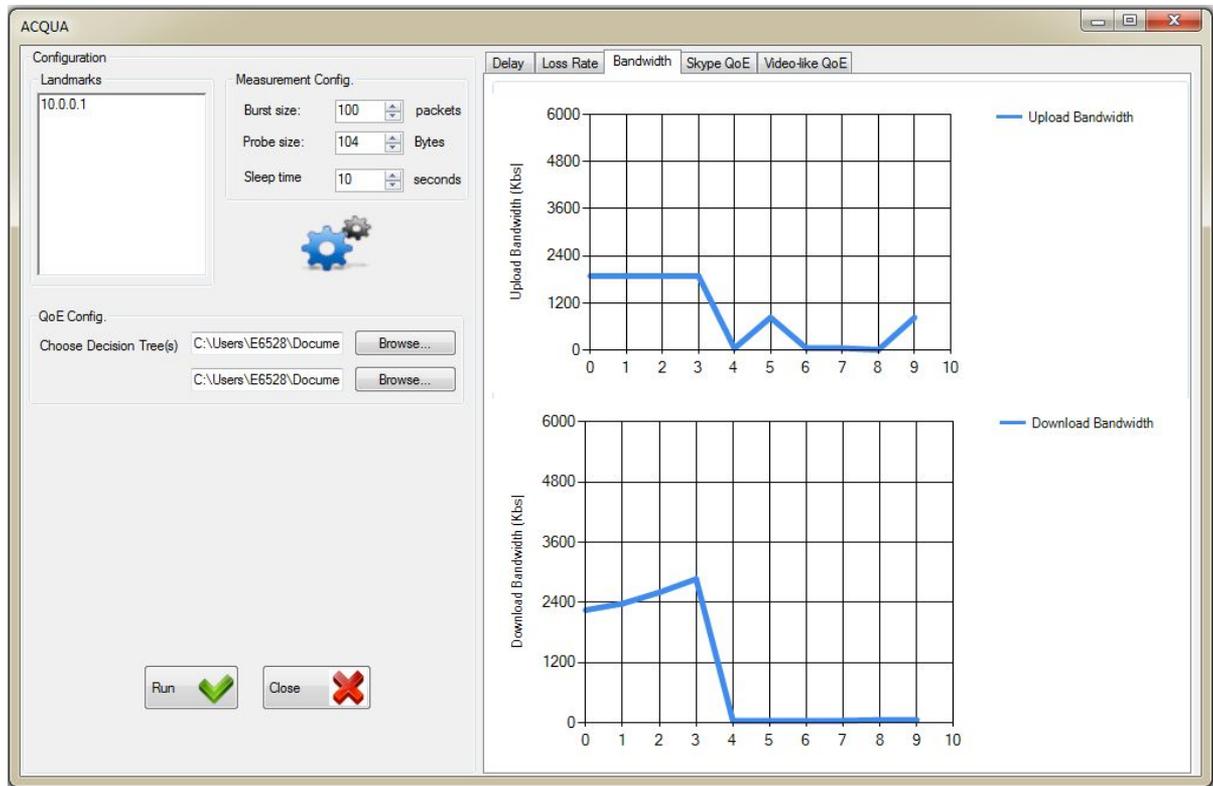
Application Startup

ACQUA Client Process

The measurement client (with GUI) is a C# application supported by Windows OSs.

Two Network QoS to QoE models are provided as XML files (embedding decision trees). The Skype model should be loaded first. The second one, and optional, is the video-like one. At the moment there is no flexibility in the model loading part.

Before starting one or more measurement server IPs should be defined as landmarks.



[Figure 1: screenshot of the GUI application]

ACQUA Server Process

The measurement server is called MeasurementServer and it is developed in Java.

To start it:

```
cd <directory of the MeasurementServer>/src
java fr.inria.acqua.ACQUAServer
```

The server will now wait for requests from the Client (deployed on Windows).

Important:

The ACQUA measurement process identifies each experiment with an ID that is not synchronized between client and server (this ID is progressive and equal to 1 in the beginning).

It is necessary to activate BOTH client and server at the same time!

(If the client is closed also the server needs to be closed in order to run another experiment).

Demo mode

The demonstration mode of the ACQUA prototype uses [DummyNet](#) to simulate network conditions between two hosts.

DummyNet is supported by most modern OS (FreeBSD, OSX, Linux, Windows).

Note: Linux kernels above version 3.2 are not well supported (in our setup we used Ubuntu 12.04 to avoid such issues).

The following instructions are relative only to the Linux version.

DummyNet management (web interface)

DummyNet is used to manipulate the network conditions.

To use DummyNet a couple of pipes has been defined, to set them up is necessary to run the script: `sudo ./ipwf-start.sh`

This will define two network pipes that will collect all incoming (pipe 10) and outgoing (pipe 11) traffic from the current host.

A static web page makes AJAX requests to a CGI server in order to manipulate the network traffic easily. This web page can be deployed on any host (be aware that DummyNet could reduce or deny completely the interaction with this remote controller by shaping the network traffic).

Reset DummyNet before starting again the application (the previous setup is still loaded and could prevent the communication between the Client and the Server).

CGI setup

`/etc/apache2/sites-available/default` contains the configuration of the Web Server (static web pages and CGI).

By default the CGI directory `/usr/lib/cgi-bin/` contains the CGI files served. It is then necessary to put here all the cgi scripts in order to configure through the web interface DummyNet.

Web pages are served from the directory `/var/www/`. The web interface (`index.html`) should be placed here (and should be available [here](#)).

Sharing a network connection from a Windows 7 host with ICS (Internet Connection Sharing)

1. Open the Network and Sharing Center
2. Click on Change Adapter Settings (sidebar)
3. Right-click on the network device with the connection to be shared (connected to Internet)
4. Select Properties
5. Go on the Sharing tab
6. enable all
7. press on Settings
8. enable all required protocols (e.g.: HTTP, HTTPS, Skype UDP, Skype TCP, ...)
9. define exceptions for ACQUA too:
 - a. UDP traffic (default 192.168.137.176:9050, 192.168.137.176:9051)
 - b. TCP traffic (default 192.168.137.176:9050, 192.168.137.176:9051)
10. Set firewall exceptions to enable ICMP (ping)
 - a. press Start
 - b. look for Run
 - c. execute `firewall.cpl` (alternatively look for Windows Firewall with Advanced Security)
 - d. click on inbound rules
 - e. add new rule (right-click or sidebar)
 - f. define a custom rule (accept everything with protocol ICMPv4)

Important: ICS will force you to use DHCP with address 192.168.137.1 as gateway. The first address assigned by the DHCP to a client is usually 192.168.137.(171-176). Windows ICS will configure also a DNS server, it is needed to set this one as DNS server on the Linux machine (see [Set DNS server on Linux host](#) - otherwise no Internet URL will be resolved correctly).

Set DNS server on Linux host

One time configuration (this configuration will be overwritten every time the host is restarted).

1. `sudo gedit /etc/resolv.conf`
2. add "nameserver 192.168.137.1" (the IP of the network interface sharing the connection)

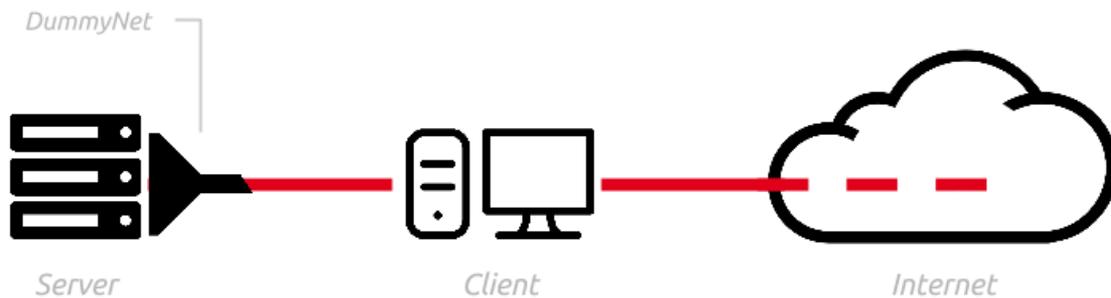
To

3. if this configuration does not hold (is overwritten) put this line in the end of the file "nohook resolv.conf"

If still not working add Firewall rules / ICS setting to allow DNS traffic on Windows

Network configuration

For the demo setup we deployed 2 machines: a Windows host (client) and a Ubuntu 12.04 (server). Both were connected by a direct wired connection. The Windows host was also connected to the Internet in order to support Skype calls and show in real time the output of both programs, Internet connection to the server was provided via ICS (Internet Connection Sharing).



[Figure 2: demo setup - network connections]

Windows

Static IP configuration is not supported while using Windows' ICS (the fixed one is 192.168.137.1).

Linux

static ip should be configured in: `/etc/network/interfaces`

This file contains the configuration of each network interface in use.

Apply network interface configurations

To update the configuration use `sudo ifup <network interface>` (e.g. eth0).

Note: the use of `sudo ifdown <network interface>` could be necessary if the interface is already up.

Important: the graphical connection manager seems to have conflicts while configuring manually the network configuration - it is recommended to disable it and proceed with the command line interface.

In case of problems (e.g.: network icon disappearing in the status bar) use `sudo service network-manager restart`.

ACQUA Desktop Prototype

Troubleshooting guide

General

Is there a video of the application?

Yes, an uncutted video is provided [here](#) and shows how our platform works in different network conditions (see [Demo Mode](#)).

How can I test the application?

This prototype comes with a “[Demo Mode](#)” that avoids a fully distributed setup. A network emulator (DummyNet) is used to test the output of the application in different network conditions and compare the predicted QoE value with the one from the Skype application.

ACQUA Platform

After rebooting a client it is not possible to start it again, why?

Both the server and the client must be started at the same time since an incremental ID is used in the traffic (not synchronized). If the synchronization between the two hosts is lost (e.g., by closing and re-opening the Client GUI) it is not possible to perform experiment. In this case the server daemon needs to be restarted.

Demo Mode

No interface are recognized when the DummyNet script is started.

All DummyNet related script must be executed using root privileges.

Limitations

The described software is a prototype and is subject to several limitations and bugs; moreover this platform is going to be unmaintained as soon as the newer version (currently under development) will be released (check the [project website](#) for updates).

Parallel measurements / flows are not supported. High experiment rates, network delays and losses value can lead to unstable behaviour and lose the synchronicity between the client and the landmark servers.

Only one client is supported by a server (single sessions are managed).

Known Bugs

Glitch in GUI, from time to time the GUI plots contains a smaller window. This issue is transitory (after a few measurement cycles the plots should return to a normal behaviour).