

# User Manual

## Flashphoner Web Call Server 3.0



# Flashphoner

## Table of Contents

<b>FLASHPHONER WEB CALL SERVER 3.0 – OPERATION PECULIARITIES .....</b>	<b>2</b>
<i>Flashphoner Web Call Server 3.0 (Flashphoner WCS) .....</i>	<i>2</i>
<i>Flashphoner WCS Technical Capacities .....</i>	<i>2</i>
<b>FLASHPHONER WCS’S AUDIENCES .....</b>	<b>2</b>
<b>FLASHPHONER WCS OPERATION PRINCIPLES .....</b>	<b>3</b>
<b>FLASHPHONER WCS TECHNICAL FEATURES .....</b>	<b>4</b>
<b>GETTING STARTED WITH FLASHPHONER WCS .....</b>	<b>5</b>
<i>Preparing the environment for installation .....</i>	<i>5</i>
<i>Step 1. Preparations prior to installation .....</i>	<i>6</i>
<i>Step 2. Program Installation .....</i>	<i>6</i>
<i>Step 3. Configuration .....</i>	<i>7</i>
<b>FLASHPHONER WCS TEST CALLS .....</b>	<b>8</b>
1. <i>Browser-to-browser audio-calls .....</i>	<i>8</i>
2. <i>Browser-to-mobile phone audio-call .....</i>	<i>9</i>
3. <i>Browser-to-softphone audio-call .....</i>	<i>10</i>
4. <i>Browser-to-browser video-call .....</i>	<i>10</i>
5. <i>Browser-to-softphone video-call .....</i>	<i>12</i>
<b>FLASHPHONER WCS DOWNLOADING .....</b>	<b>13</b>
<b>GLOSSARY .....</b>	<b>14</b>

Adobe, Flash and Flex are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Intel, Intel Xeon, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Red Hat is a trademark or registered trademark of Red Hat, Inc. in the United States and other countries. Windows Server is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

This document provides detailed information on the Flashphoner Web Call Server software and a step-by-step guide of its configuration.

## Flashphoner Web Call Server 3.0 – Operation Peculiarities

### Flashphoner Web Call Server 3.0 (Flashphoner WCS)

Flashphoner WCS is server software which allows establishing a real-time connection between Adobe Flash apps, WebRTC supported browser and any VoIP server.

You also get the possibility to create and configure a customized flash-softphone, "Call from the website" and "Web-phone" javascript/html-widgets, as well as all tools necessary for of WebRTC-VoIP or Flash-VoIP connection establishment directly from your browser. The product allows for implementation of such features as audio conferencing, video conferencing, or a channel for online radio calls receiving.

### Flashphoner WCS Technical Capacities

The Flashphoner WCS software has been designed with open-source Flash/AS3 and Javascript/CSS codes. As a result, it is possible to implement customized graphical interfaces for any website.

- WebRTC technology and RTMFP, SIP protocols
- Speex Wideband, G.711, G.729, Opus WebRTC audio codecs support
- H.263, H.264, Sorenson Spark video codecs support
- DTMF rfc2833 and SIP INFO support
- Text messages sending and receiving via SIP MESSAGE

## Flashphoner WCS's Audiences

Flashphoner WCS allows performing various tasks in order to achieve personal, as well as corporate objectives. For convenience, the tasks have been divided into the following groups; each of these tasks can be performed for your personal ends.

### Corporate Projects

- Integration into Call-and Contact-Centers;
- Integration into a virtual PBX;
- Integration into corporate CRM and ERP systems;
- Configuration of inbound and outbound IVR systems;
- Audio- and video-conferencing.

### Marketing and Advertising

- "Call from the website" widget;
- "Free call" button used in organizations electronic catalogues for direct contact with the organization.

### Web Services

- Web-phones on a website or in a user's personal account;
- Adobe AIR-softphones for computers;
- Phone-applications for social networks;
- Voice and video mail.

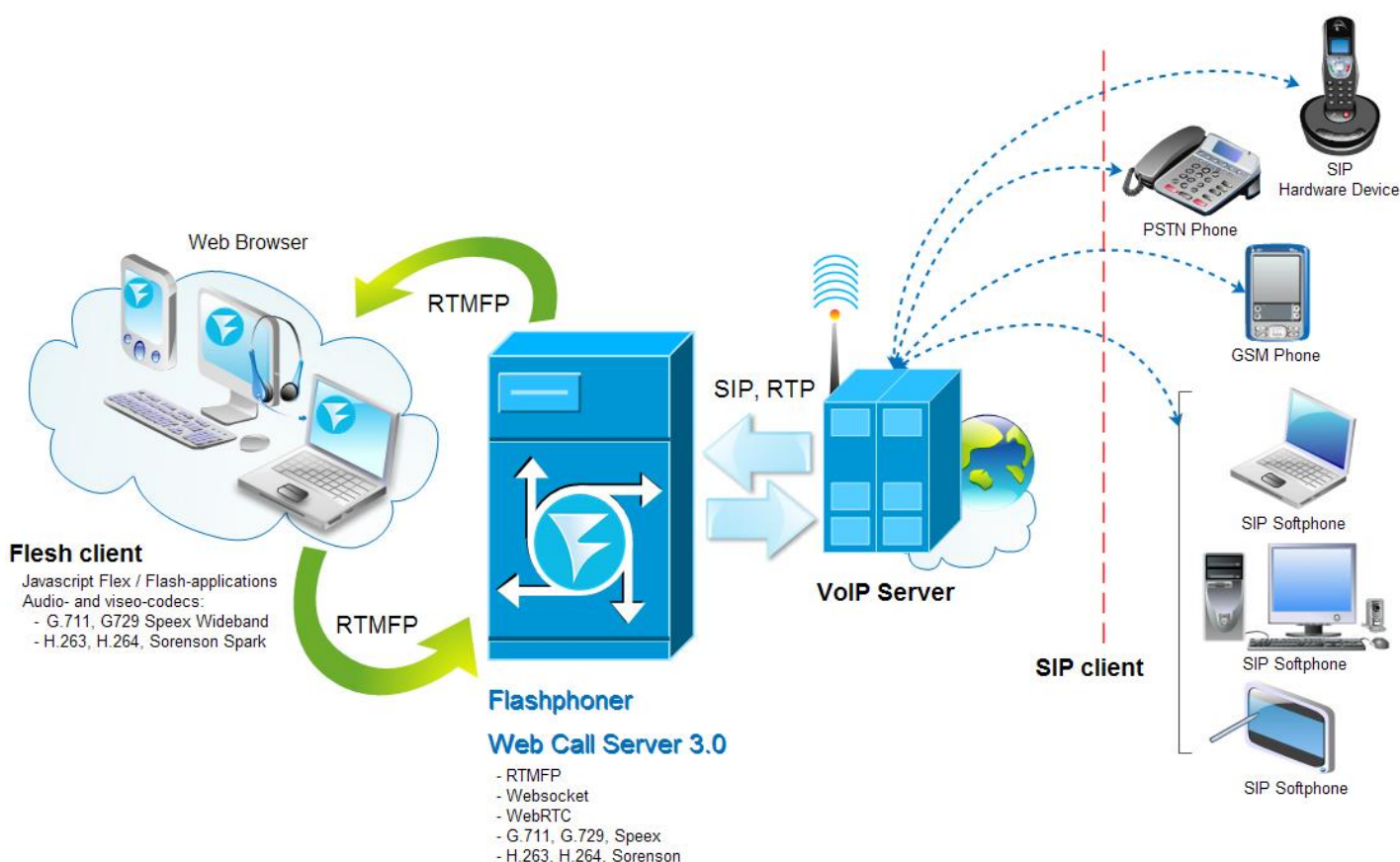
### On-line streaming

- E-learning – webinars and lectures via a phone;
- Web-TV – sending comments and messages over the phone;
- Web-TV/Radiogames – mobile phone calls of the participants.

## Flashphoner WCS Operation Principles

Flashphoner WCS functions in the following way:

**Flash-client ↔ Flashphoner WCS ↔ VoIP server ↔ SIP client**



### Flash-client

- The Flash-client is a WebRTC/Flex/Flash application. It can be a web-phone, a “Call from the website” widget, an on-line banner, a conference platform or any other application;
- This level uses RTMFP and WebRTC for streaming.
- The following audio- and video-codecs are supported: G.711, G, 729, Speex Wideband, H.263, H.264, and Sorenson Spark;
- The Flashphoner WCS basic configuration provides for the flash-client, however, you are free to implement your own design as an application.

### Flashphoner WCS Platform

- Flash-client data is received and communicated to the VoIP-server;
- Flash-client connection is implemented through RTMFP or WebRTC;

- VoIP-server connection is implemented through SIP;
- The following audio- and video-codecs are supported: G.711, G, 729, Speex Wideband, H.263, H.264, and Sorenson Spark

### VoIP-server

- The VoIP-server can be implemented on a conventional or a virtual IP PBX:
  - Virtual PBX (software-based platform) – Asterisk, Trix-box, Yate, CommunigatePro, OpenSIPs, Cisco Call Manager, etc;
  - Hardware-based PBX – Avaya, Cisco, etc.;
- Flashphoner WCS transmits data to the VoIP-server which further communicates it to the SIP-client;
  - The VoIP-server can be installed on the same physical server with Flashphoner WCS and can be local or leased from a VoIP-operator.

Correct server configuration is crucial (i.e. allowing for SIP accounts registration, etc.)

### SIP-client

- In order to ensure outbound connection to the Flash-client, the SIP-account settings should be configured (username/password provided by the VoIP-provider; the provider server; open port 5060) and audio- and video-codecs similar to the Flash-client (G.711, G,729, Speex Wideband, H.263, H.264, Sorenson Spark) should be supported.
  - In order to ensure receiving of incoming calls, the phone shall support audio- and video-codecs (G.711, G,729, Speex Wideband, H.263, H.264, Sorenson Spark) and shall be conventional, software-based, or hardware-based.
    - Conventional – land-line or mobile phones;
    - Software-based phones – PortGo, Bria, EyeBeam, Linphone, Mirial Softphone, Zoiper, Ekiga, etc.;
    - Hardware-based – Cisco, Tanderberg, SNOM, Grandstream, ipDialog, etc.

## Flashphoner WCS Technical Features

The “Technical Features” section contains information on minimum hardware and system requirements, as well as on the supported Flashphoner WCS specifications.

Environmental requirements	
<b>Production hardware requirements</b>	<ul style="list-style-type: none"> <li>• Quad Core CPU 2.5Ghz (2 Quad-Core processors advisable)</li> <li>• RAM 4 GB or higher (8-16GB advisable)</li> <li>• 100 Mbps Ethernet-board</li> </ul>
<b>System requirements</b>	<ul style="list-style-type: none"> <li>• Linux x86_64, x86 (CentOS, Debian, Ubuntu, RedHat, Fedora)</li> <li>• Java Development Kit (JDK 1.7+)</li> <li>• Hardware platform x86-64</li> </ul>
Supported specifications	
<b>SIP 2.0</b>	<a href="#">rfc3261</a>

<b>RTP</b>	<a href="#">rfc3550</a>
<b>SDP</b>	<a href="#">rfc4566</a>
<b>RTMFP</b>	RTMFP (Real-Time Media Flow Protocol)
<b>DTMF</b>	SIP INFO, <a href="#">RFC2833</a>
<b>Transfer, Hold</b>	<a href="#">RFC 3515</a>
<b>Instant Messaging</b>	<a href="#">RFC 3428</a>
<b>Codecs</b>	<b>Audio</b> <ul style="list-style-type: none"> <li>- Speex Wideband 16Khz</li> <li>- G.711 alaw, ulaw</li> <li>- Opus WebRTC codec</li> <li>- G.729A</li> </ul> <b>Video</b> <ul style="list-style-type: none"> <li>- H.263, H.263+ CIF (352x288)</li> <li>- H.264 Baseline 3,0 (up to 720p)</li> <li>- Sorenson Spark (Flash to Flash calls)</li> </ul>
<b>Formats</b>	<b>Audio</b> <ul style="list-style-type: none"> <li>- annexb = no</li> <li>- ptime: 20</li> <li>- ptime: 30</li> </ul>
<b>Flash Player, WebRTC</b>	Adobe Flash Player 11 or higher, Latest Chrome and Firefox
<b>Client platform</b>	Flex/Javascript
<b>Server platform</b>	Java, C

## Getting Started with Flashphoner WCS

We recommend you to make a test call using the [Flashphoner WCS softphone](#). Use the data obtained at registration with any VoIP-connection SIP-operator. Log in and make some test calls. The detailed examples of test calls are provided in section "Flashphoner WCS Test Calls".

Now you need to prepare the environment in order to configure and test the software.

### Preparing the environment for installation

To prepare the environment for installation, you should have root access to the server (SSH connection) and the software necessary for Flashphoner WCS operation installed.

1. Download and install the latest JDK version (1.7 or higher)

#### Peculiarities of JVM, JDK configuration and testing of

If you use a x86\_64 system, use a 64-bit JVM.

This can be checked in the following way:

```
$uname -a
```

Result:

```
2.6.18-164.el5xen #1 SMP Thu Sep 3 04:03:03 EDT 2009 x86_64 x86_64 x86_64 GNU/Linux
```

```
$java -version
```

Result:

```
java version "1.6.0_21"
```

```
Java(TM) SE Runtime Environment (1.6.0_21-b06)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 17.0-b16, mixed mode)
```

If the installed JVM does not support the 64-bit mode, download the latest JDK installation file from the <http://oracle.com> web-site, uninstall your current JDK, and install the latest JDK. If you

have problems during the installation, please refer to section "[Possible problems during the installation of JDK](#)".

Once you have checked and/or installed JDK 1.7, please connect to the server via SSH and perform the following actions in command line:

2. Create a symlink using the ln command:  
**`$ln -sf /usr/java/default/bin/java /usr/bin`**

Once you've prepared the environment, install and configure Flashphoner WCS. Launching the Flashphoner Web Call Server platform requires three steps:

### Step 1. Preparations prior to installation

Make sure that you have an active account in any VoIP-service and the SIP configurations (registered client username/password, provider Proxy Server and open port 5060) before installation of Flashphoner Web Call Server. You should have a license key sent to the specified e-mail after filling form on the Flashphoner.com website.

The latest JDK (Java Development Kit) version should be installed on your Linux-server. You can download it for free from oracle.com.

The recommended architecture of the Linux server is x86\_64.

### Step 2. Program Installation

#### Server Side

1. Download and unpack the downloaded archive of [Flashphoner-Web-Call-Server-3.0.X.tar.gz](#)

2. Run the **install.sh** installation script

During the installation, you will need to enter your server IP-address and the license key. If for some reason you did not receive the key to the e-mail specified in the application, you can get it [here](#).



#### **NOTE**

*Make sure that you've entered a correct ip and ip\_local during the installation. Check the settings of the flashphoner.properties configuration file prior to working with the Flashphoner WCS software.*

#### **ip and ip\_local settings**

The following ip and ip\_local settings should be configured during the Flashphoner WCS installation.

- **ip** – external IP-address of the server
- **ip\_local** – local IP-address of the server

There are three basic options for setting ip and ip\_local:

1. The Linux server with Flashphoner WCS has a local ip address 192.168.1.5 and the VoIP server is in the same local network and has ip address 192.168.1.6.

In this case, you need to specify: ip = 192.168.1.5 and ip\_local = 192.168.1.5

2. The Linux server with Flashphoner WCS has a local ip address 192.168.1.5 and the VoIP server is in the global network of an external SIP-provider, for example 212.53.40.40.

In this case, you need to determine the external address of your server and enter it in the ip properties. To determine the external ip address, you can use a service like <http://www.whatismyip.com>.

Suppose your external ip address: 87.224.223.55.

In this case, you should specify: ip = 87.224.223.55 ip\_local = 192.168.1.5

3. Linux server and a VoIP server are both in the global network.

Suppose you are using an external leased or own server, which is serviced by data center and its open static ip address is 87.224.223.55.

In this case, you must specify the ip = 87.224.223.55 ip\_local = 87.224.223.55

To find out the ip address of your server, run the \$ ifconfig command.

 **NOTE**

You can change the parameters of ip and ip\_local in the flashphoner.properties configuration file after installation. See the [Configuration Section](#)

*Finally, the data shall be entered in accordance with one of the options as follows:*

Environment	ip	ip_local
Server is in the local network	192.168.1.5	192.168.1.5
One of the servers is local, the other is global	87.224.223.55	192.168.1.5
Both servers are in the global network	87.224.223.55	87.224.223.55

Client Side

1. Download and unpack [Flashphoner-Web-Call-Client-3.0.X.tar.gz](#), copy it to your Linux-server directory, for example, /var / www / html / Flashphoner-Web-Call-Client.

Make sure that the ip :/ Flashphoner-Web-Call-Client/PhoneJS.html page opens correctly in your browser.

2. Change the path to the RTMFP server in flashphoner.xml. It should refer to your server, for example, **rtmfp :/ / xxx.xxx.xxx.xxx: 1935**. Then open ip :/ Flashphoner-Web-Call-Client/flashphoner.xml in the browser and make sure the path is set correctly.

### Step 3. Configuration

1. Turn off the firewall or open the 1935 UDP port and UDP ports ranges [port\_from, port\_to] UDP and [media\_port\_from, media\_port\_to]. Ranges [30000-31000] for signaling traffic and [31001-32000] for media traffic are used by default. You can change these settings in configuration file flashphoner.properties.

2. Run the Web Call Server using command: **./startup.sh**

You can find server logs in **logs/flashphoner.log**

3. Open the webpage PhoneJS.html in your browser.

**Now you can make test calls!**

## Flashphoner WCS Test Calls

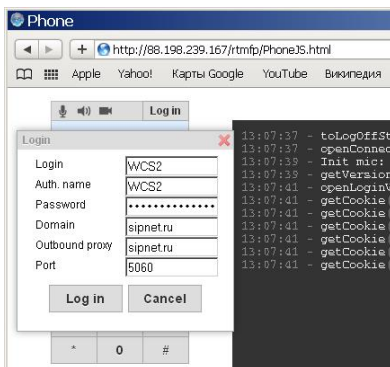
To test Flashphoner WCS functioning with Flash and SIP clients, you need two funded accounts of any VoIP-connection SIP-operator (the amount of should cover at least 1-minute call to a GSM-operated phone)

### 1. Browser-to-browser audio-calls

You need 2 VoIP-provider SIP-accounts to make this type of test calls.

Accounts WCS1 and WCS2, provided in the example, have been obtained from the SIPNET VoIP-provider. After account settings have been received from SIPNET, enter them in the [Flashphoner WCS browser softphone](#) by clicking “Login in”.

The fields shall be filled as shown at fig.1:



**Login:** **WCS1** (or **WCS2**)

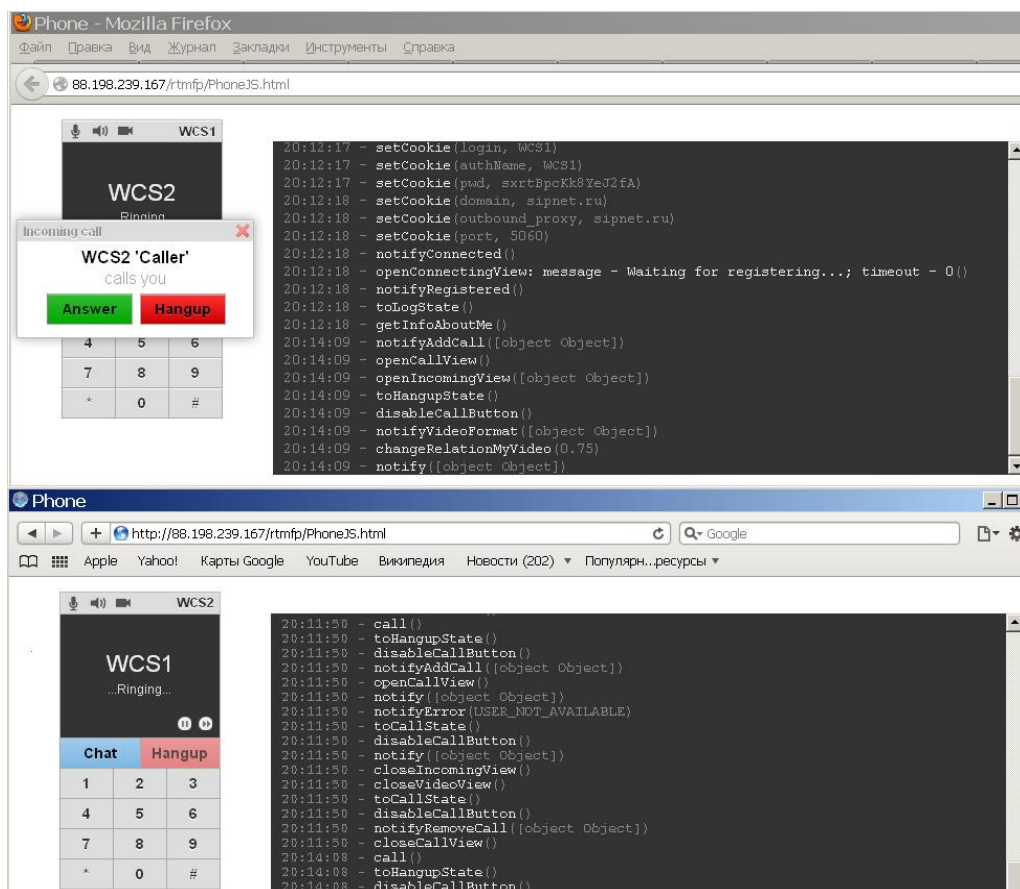
**Password:** \*\*\*\*\* (password of the account registered with the VoIP-provider)

**Domain:** **sipnet.ru** (provided by the VoIP-provider)

**Port:** **5060**

The other fields are filled in automatically.

**Fig.1** – Configuring Flashphoner WCS browser-softphone



**Fig. 2** – “Browser-to-browser” audio-call



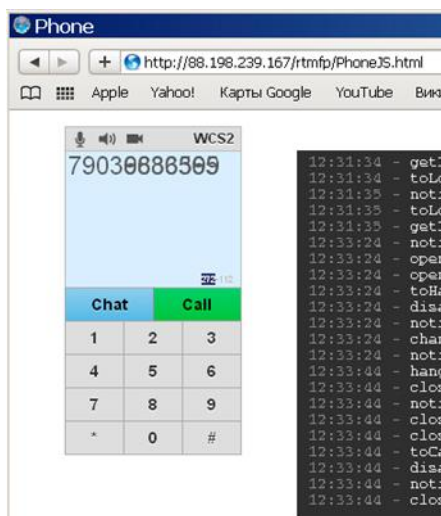
The example made use of two different browsers installed on one computer. In order to test the quality of connection we suggest entering the settings to Flashphoner WCS browser-softphones run on different devices (PC, laptop, tablet).

Once the settings have been entered, calls between browser softphones can be made. Fig.2 illustrates a call from one browser (WCS1) to the account registered in another browser (WCS2).

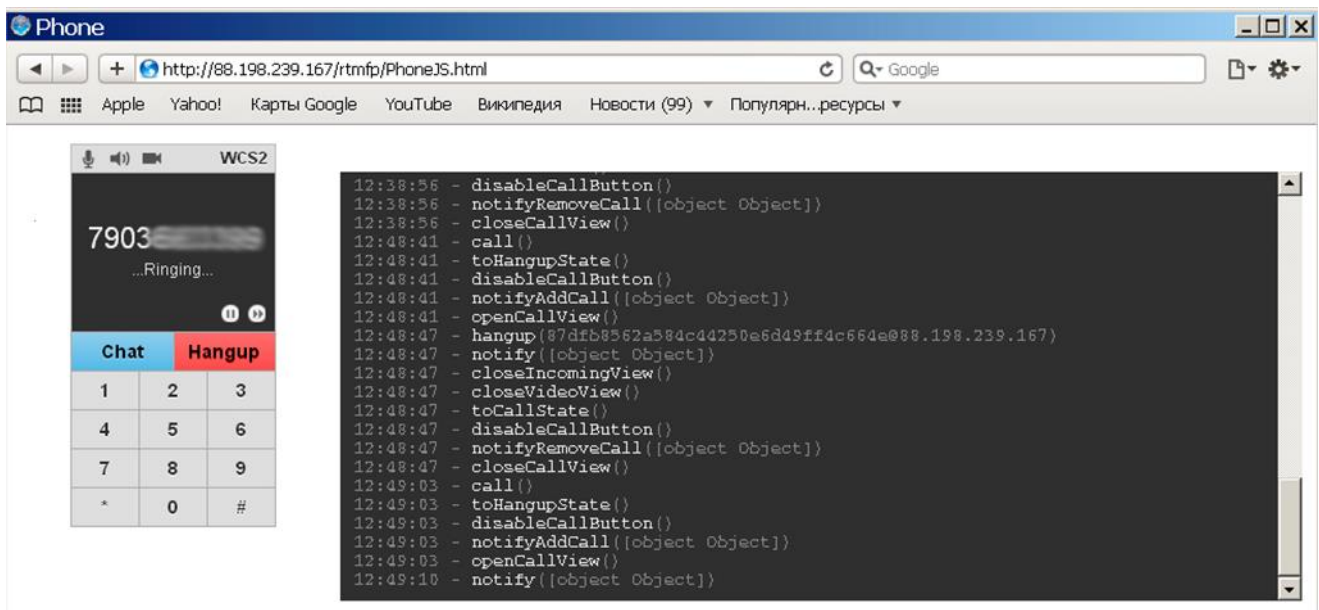
## 2. Browser-to-mobile phone audio-call

The configuration of Flashphoner WCS browser softphone is similar to that described in clause 1 of this section.

Once the settings have been introduced, you can call both to mobile and land-line phones, in accordance with the rules of international dialing with no spaces and other punctuation. Fig. 3 and 4 illustrate the dialing format and connection with a subscriber.



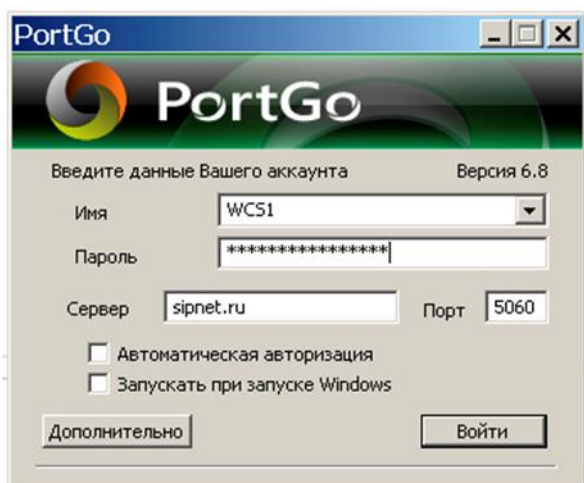
**Fig.3** – Entering a Russian mobile operator number in the international format.



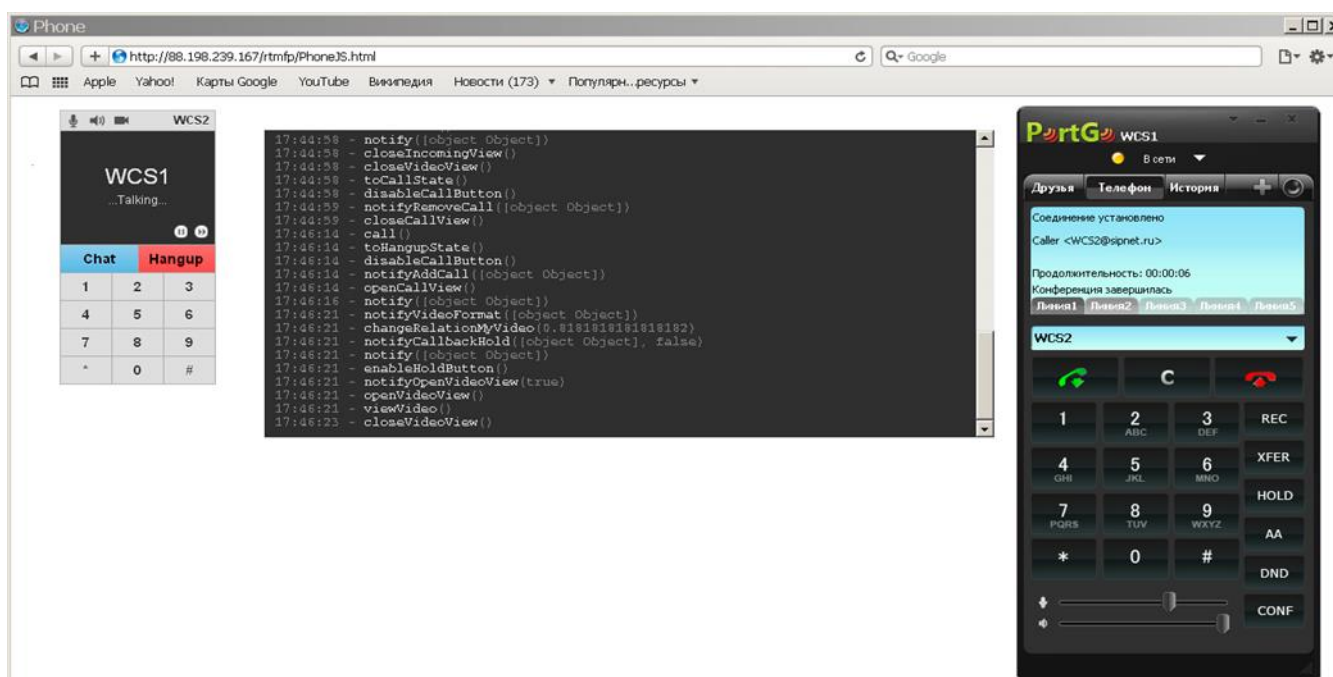
**Fig.4** – Connection to a subscriber's mobile phone via Flashphoner WCS browser softphone

### 3. Browser-to-softphone audio-call

When testing this type of connection, you need two VoIP-provider accounts. Configure Flashphoner WCS browser softphone in the way described in clause 1 of the section. Then select and install a softphone, PortGo in this particular case. Once the softphone is installed, enter your parameters as shown on Fig. 5. The call is made to the login. Once these steps are taken, your softphone is ready for use and making test calls.



*Fig.5 –Softphone settings*

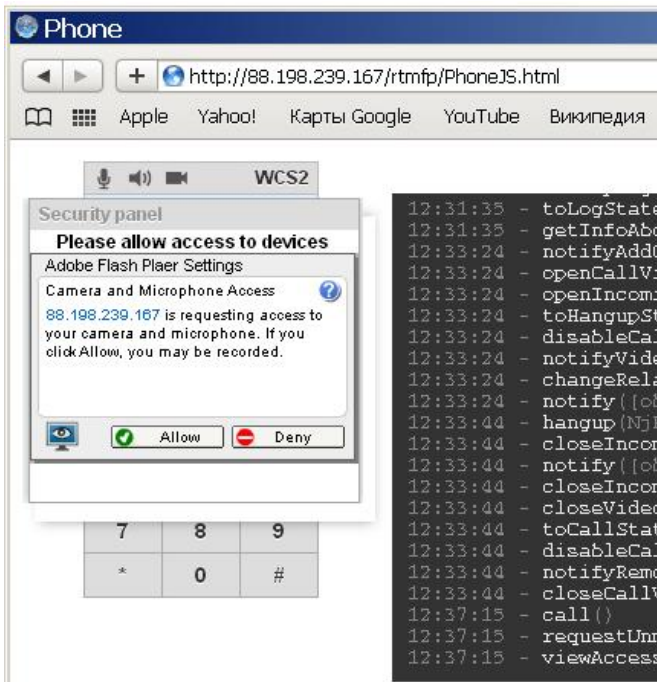


*Fig.6 Conversational mode during a browser-to-softphone audio-call*

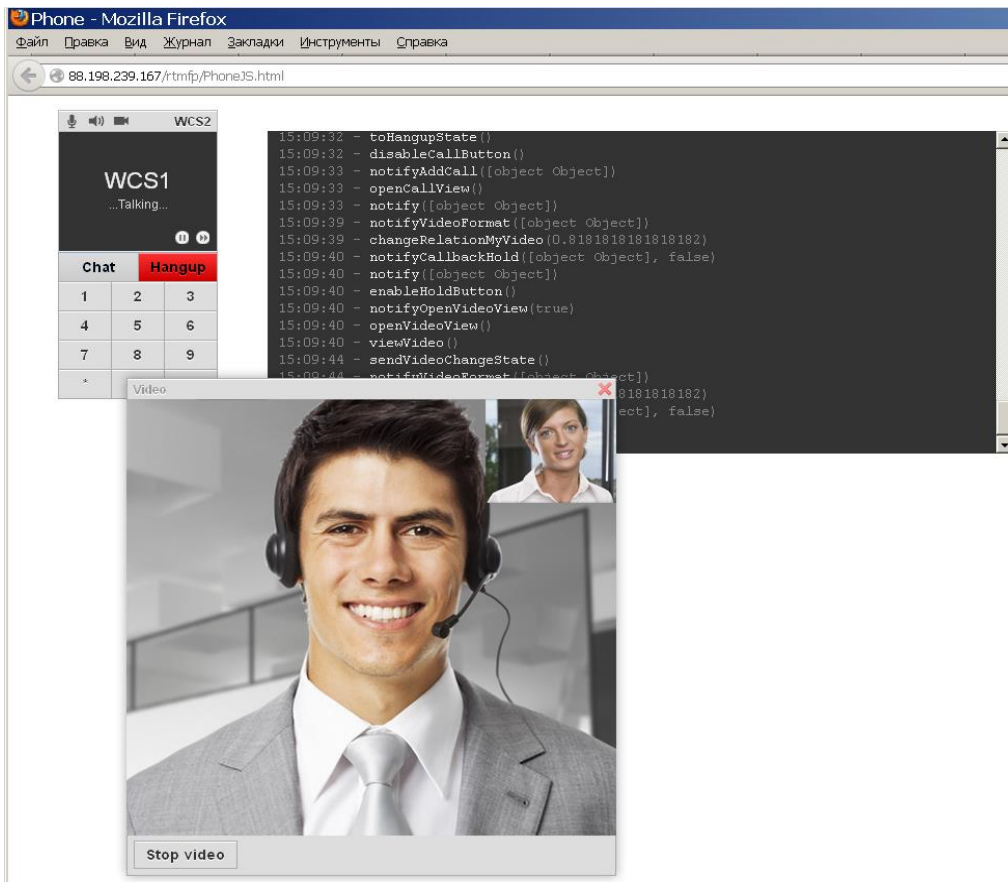
### 4. Browser-to-browser video-call

In order to make test browser-browser calls, you need to configure two Flashphoner WCS browser softphone accounts on two different devices (PC, laptop, tablet). Flashphoner WCS browser

softphone is configured in the way described in clause 1 of this section. Test calls are made to the account's username. When making a call, allow access to the web-camera and the microphone by clicking "Enable" in the popup window while connecting to the subscriber, see. Fig. 7.



**Fig.7 – Allowing a browser-to-browser video-call**

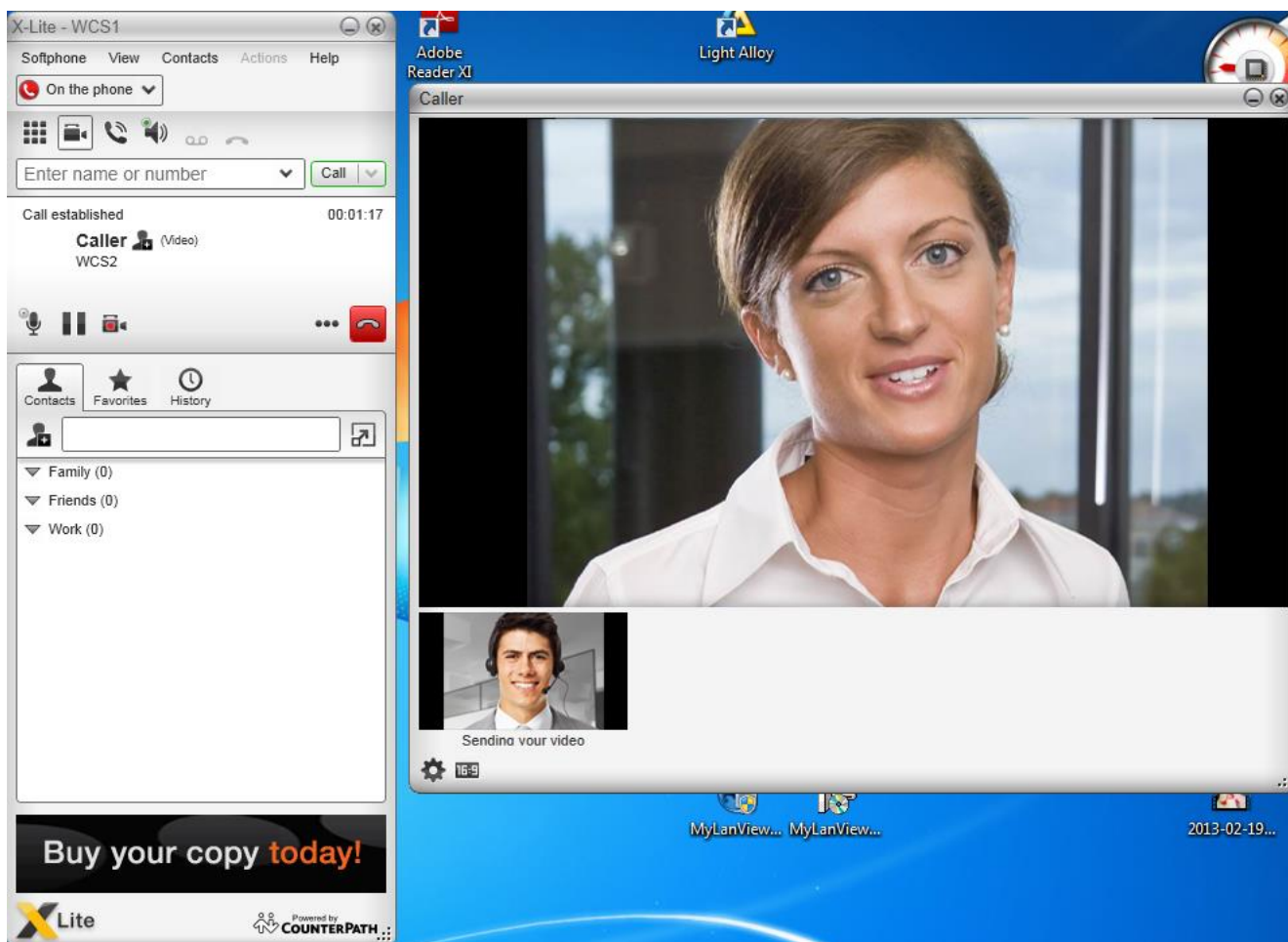


**Fig.8 – Browser-to-browser video-call**

## 5. Browser-to-softphone video-call

Flashphoner WCS browser softphone and a software-based phone should be configured as shown in clauses 1 and 3 respectively. The example makes use of the XLite software-based phone. Two accounts on a software-based and a browser phone were connected.

During a call, access to the web-camera and the microphone should be allowed by clicking “Enable” in the popup window during connection to the subscriber. In order to get accurate video streaming, click “Send video” in the browser softphone. The softphone window shall display the video-call. See fig.9.



**Fig. 9** – Softphone-to-browser video-call

If you want to open the browser page locally without using a Web-server, please refer to the FAQ section “How to use the SWF client locally” on the Flashphoner site.

If you have any questions or problems during the installation, our support service will find a solution. Send a request or contact us.

## Flashphoner WCS downloads

Download and install the Flashphoner WCS software.

### **Flashphoner WCS Server-side**

<http://flashphoner.com/downloads/builds/WCS/3.0/x8664/>

### **Flashphoner WCS Client**

[http://flashphoner.com/downloads/builds/flashphoner\\_client/wcs-3.0/](http://flashphoner.com/downloads/builds/flashphoner_client/wcs-3.0/)

Feel free to contact us with any questions. You can always find our contact info at the Flashphoner.com website.

## Glossary

**Adobe AIR (Adobe Integrated Runtime)** is a cross-platform run-time system developed by Adobe Systems for building Rich Internet applications (RIA) using Adobe Flash, Apache Flex (formerly Adobe Flex), HTML, and Ajax, that can be run as desktop applications or on mobile devices. The runtime supports installable applications on Windows, Mac OS and some mobile operating systems such as BlackBerry Tablet OS, iOS and Android. It also originally supported Linux, but that support was discontinued.

**Asterisk** is a software implementation of a telephone private branch exchange (PBX); it was created in 1999 by Mark Spencer. Like any PBX, it allows attached telephones to make calls to one another, and to connect to other telephone services, such as the public switched telephone network (PSTN) and Voice over Internet Protocol (VoIP) services. Its name comes from the asterisk symbol, \*.

Asterisk is released under a dual license model, using the GNU General Public License (GPL) as a free software license and a proprietary software license to permit licensees to distribute proprietary, unpublished system components.

Originally designed for Linux, Asterisk also runs on a variety of different operating systems including NetBSD, OpenBSD, FreeBSD, Mac OS X, and Solaris. A port to Microsoft Windows is known as AsteriskWin32. Asterisk is small enough to run in an embedded environment like Customer-premises equipment-hardware running OpenWrt.

**DTMF** (Dual-tone multi-frequency signaling) is used for telecommunication signaling over analog telephone lines in the voice-frequency band between telephone handsets and other communications devices and the switching center. The version of DTMF that is used in push-button telephones for tone dialing is known as Touch-Tone. DTMF is standardized by ITU-T Recommendation Q.23. It is also known in the UK as MF4. Other multi-frequency systems are used for internal signaling within the telephone network.

Introduced by AT&T in 1963, the Touch-Tone system using the telephone keypad gradually replaced the use of rotary dial and has become the industry standard for landline service.

**GNU Compiler Collection (GCC)** is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU toolchain. The Free Software Foundation (FSF) distributes GCC under the GNU General Public License (GNU GPL). GCC has played an important role in the growth of free software, as both a tool and an example. Although it runs the risk of losing ground to the more liberally licensed, cleaner written, Clang Compiler if current momentum of that project continues.

Originally named the GNU C Compiler, because it only handled the C programming language, GCC 1.0 was released in 1987 and the compiler was extended to compile C++ in December of that year. Front ends were later developed for Objective-C, Objective-C++, Fortran, Java, Ada, and Go among others. GCC is also available for most embedded platforms, including Symbian (called gcce), AMCC, and Freescale Power Architecture-based chips.

**Instant messaging (IM)** is a form of communication over the Internet that offers quick transmission of text-based messages from sender to receiver. In push mode between two or more people using personal computers or other devices, along with shared clients, instant messaging basically offers real-time direct written language-based online chat. The user's text is conveyed over a network, such as the Internet. It may address point-to-point communications as well as multicast communications

from one sender to many receivers. More advanced instant messaging allows enhanced modes of communication, such as live voice or video calling, video chat and inclusion of hyperlinks to media.

**Java Development Kit (JDK)** is an implementation of either one of the Java SE, Java EE or Java ME platforms[1] released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows. Since the introduction of Java platform, it has been by far the most widely used Software Development Kit (SDK). On 17 November 2006, Sun announced that it would be released under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007, when Sun contributed the source code to the OpenJDK.

**Java virtual machine (JVM)** is a virtual machine that can execute Java bytecode. It is the code execution component of the Java platform. Sun Microsystems has stated that there are over 5.5 billion JVM-enabled devices.

A Java virtual machine is a program which executes certain other programs, namely those containing Java bytecode instructions. JVMs are most often implemented to run on an existing operating system, but can also be implemented to run directly on hardware. A JVM provides a run-time environment in which A JVM is distributed along with Java Class Library, a set of standard class libraries (in Java bytecode) that implement the Java application programming interface (API). These libraries, bundled together with the JVM, form the Java Runtime Environment (JRE).

**Make** is a utility that automatically builds executable programs and libraries from source code by reading files called makefiles which specify how to derive the target program. Though integrated development environments and language-specific compiler features can also be used to manage a build process, Make remains widely used, especially in Unix.

**Secure Real-Time Media Flow Protocol (RTMFP)** is a proprietary protocol suite developed by Adobe Systems for encrypted, efficient multimedia delivery through both client-server and peer-to-peer models over the Internet.

RTMFP will reduce the bandwidth costs for direct, live, real-time communication solutions, such as audio and video chat and multi-player games. Because RTMFP flows data between the end-user clients and not the server, bandwidth is not being used at the server so solutions are less expensive to scale. RTMFP also increases the speed of delivery through the use of UDP. UDP is a more efficient (but less reliable) way to send video and audio data over the Internet that reduces the penalties associated with missing, dropped, or out of order packets. RTMFP has two features that may help to mitigate the effects of connection errors.

**Real Time Messaging Protocol (RTMP)** was initially a proprietary protocol developed by Macromedia for streaming audio, video and data over the Internet, between a Flash player and a server. Macromedia is now owned by Adobe, which has released an incomplete version of the specification of the protocol for public use.

RTMPT which is encapsulated within HTTP requests to traverse firewalls. RTMPT is frequently found utilizing cleartext requests on TCP ports 80 and 443 to bypass most corporate traffic filtering. The encapsulated session may carry plain RTMP, RTMPS, or RTMPE packets within.

While the primary motivation for RTMP was to be a protocol for playing Flash video, it is also used in some other applications, such as the Adobe LiveCycle Data Services ES.

**Real-time Transport Protocol (RTP)** defines a standardized packet format for delivering audio and video over IP networks. RTP is used extensively in communication and entertainment systems that involve streaming media, such as telephony, video teleconference applications, television services and web-based push-to-talk features.

RTP is used in conjunction with the RTP Control Protocol (RTCP). While RTP carries the media streams (e.g., audio and video), RTCP is used to monitor transmission statistics and quality of service (QoS) and aids synchronization of multiple streams. RTP is originated and received on even port numbers and the associated RTCP communication uses the next higher odd port number.

RTP is one of the technical foundations of Voice over IP and in this context is often used in conjunction with a signaling protocol which assists in setting up connections across the network.

RTP was developed by the Audio-Video Transport Working Group of the Internet Engineering Task Force (IETF) and first published in 1996 as RFC 1889, superseded by RFC 3550 in 2003.

**The Session Initiation Protocol (SIP)** is an IETF-defined signaling protocol widely used for controlling communication sessions such as voice and video calls over Internet Protocol (IP). The protocol can be used for creating, modifying and terminating two-party (unicast) or multiparty (multicast) sessions. Sessions may consist of one or several media streams.

Other SIP applications include video conferencing, streaming multimedia distribution, instant messaging, presence information, file transfer and online games[citation needed].

SIP is an application layer protocol designed to be independent of the underlying transport layer; it can run on Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Stream Control Transmission Protocol (SCTP). It is a text-based protocol, incorporating many elements of the Hypertext Transfer Protocol (HTTP) and the Simple Mail Transfer Protocol (SMTP).

**The Transmission Control Protocol (TCP)** is one of the two original core protocols of the Internet protocol suite (IP), and is so ubiquitous that the entire suite is often called TCP/IP. TCP provides reliable, ordered, error-checked delivery of a stream of octets between programs running on computers connected to an intranet or the public Internet.

Browsers use it when they connect to servers on the World Wide Web sites, and it is used to accurately deliver email and transfer files from one location to another. Applications that do not require the reliability of a TCP connection may instead use the connectionless User Datagram Protocol (UDP), which emphasizes low-overhead operation and reduced latency rather than error checking and delivery validation.

**The User Datagram Protocol (UDP)** is one of the core members of the Internet protocol suite, the set of network protocols used for the Internet. With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network without prior communications to set up special transmission channels or data paths. The protocol was designed by David P. Reed in 1980 and formally defined in RFC 768.

UDP uses a simple transmission model with a minimum of protocol mechanism. It has no handshaking dialogues, and thus exposes any unreliability of the underlying network protocol to the user's program. As this is normally IP over unreliable media, there is no guarantee of delivery, ordering or duplicate protection. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram.