

**ACADEMIA DE STUDII ECONOMICE - București**  
**Bucharest University Of Economic Studies**  
**FACULTY OF BUSINESS ADMINISTRATION**  
(Facultatea de Administrare a Afacerilor cu predare în limbi străine)

**Technologies on e-Business**  
-  
**The Architecture of Internet**

**By: Professor Vasile AVRAM, PhD**  
- suport de curs destinat studenților de la secția engleză - licența -  
(Lecture notes for 2<sup>nd</sup> year students of English division)  
- anul II - Zi -  
(last update: October, 2013)

**București 2013**

COPYRIGHT® 2008-2015  
All rights reserved to the author Vasile AVRAM.

# 1 Internet – ARCHITECTURE, OFFERED SERVICES, COMMUNICATION AND NAVIGATION

## 1.1 How WANs (and Internet) are organized

Arguably, the today's Internet, is the largest engineered open system ever created by mankind, with hundreds of millions of connected computers, communications links, and switches together with a “web of sites” used to disseminate data, information, knowledge, tools, applications etc. Hundreds of millions of users connected intermittently via cell phones, tablets, PDAs, and devices such as sensors, webcams, games consoles, picture frames etc.

The WANs (and the most general one, Internet) are composed from (figure 1.1):

- at the lower level LANs, MANs etc or in other words **sub-networks** ( $S_{ij}$ );
- at the next up the sub-networks are linked together, by using inter-network devices, in **areas** ( $A_{k,l}$ );
- the areas are linked together, by means of routers, into **domains** ( $D_m$ );
- all connected domains (by means of routers and using a packet or circuit switching transmission technology) form the **WAN**.

The Internet is not a single network, but a worldwide collection of loosely connected networks that are accessible by individual computer hosts in a variety of ways, including gateways, routers, dial-up connections, and Internet Service Providers (ISP). The Internet is easily accessible to anyone with a computer and a network connection. Individuals and organizations worldwide can reach any point on the network without regard to national or geographic boundaries or time of day.

The Internet community consists of individuals, business, and a variety of organizations locate throughout the world. We can consider that most members of the Internet community fall into one or more from the following groups:

- Users – are people that mainly use Internet to retrieve content or perform online activities;
- Internet Service Providers (ISPs) – are organizations (business or other) that provide Internet access to others (typically on a subscription basis or for a fee);
- Internet Content Providers – these supply the information that is available through Internet (photos, movies, books, articles, opinions, software for download etc);
- Application Service Providers (ASPs) and Web Services – are companies that distribute Web-based software services to customers over the Internet;
- Infrastructure Companies – are the enterprises that own or operate the paths or the “roadways” of Internet (such as backbones and networks connected to it);

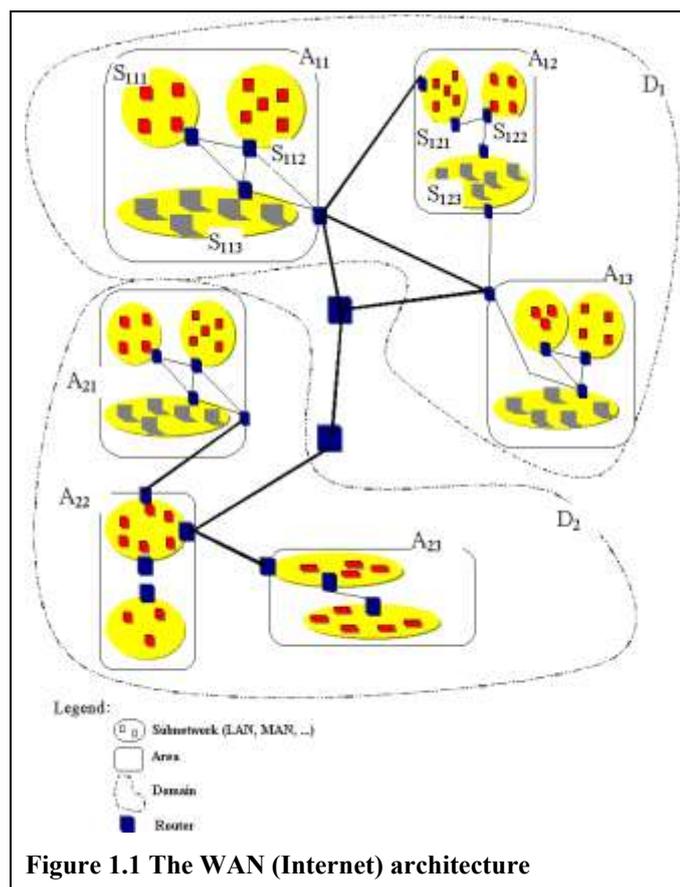


Figure 1.1 The WAN (Internet) architecture

- Hardware and Software Companies – they deliver a wide variety of hardware and software products used with the Internet and Internet activities;
- Government – the governments in different countries or areas can impose by law and restrict the usage of Internet as a vector for dissemination of different kind of content;
- Internet specific organizations – are the organizations that defines the communication and connectivity standards (such as Internet Engineering Task Force – IETF), Internet Corporation for Assigned Names and Numbers (ICANN) or the definition of languages and technologies used in WWW (such as World Wide Web Consortium – W3C) etc.

The main reason most people buy a modem—or an entire PC, for that matter—is to connect to the Internet. In the following tables is shown the number of people using Internet in the world and the position of Europe and Romania (before accessing EU) in these statistics for 2005:

a) the Internet usage at world level by regions

Internet usage						
Regions	Population ( 2005 Estimated)	% percent from total population	Internet Usage	Grows 2000-2005	% Population ( Penetration )	% usage at world level
Africa	896,721,874	14.0 %	23,867,500	428.7 %	2.7 %	2.5 %
Asia	3,622,994,130	56.4 %	327,066,713	186.1 %	9.0 %	34.2 %
Europe	731,018,523	11.4 %	273,262,955	165.1 %	37.4 %	28.5 %
Middle East	260,814,179	4.1 %	21,422,500	305.4 %	8.2 %	2.2 %
North America	328,387,059	5.1 %	223,779,183	107.0 %	68.1 %	23.4 %
Latin America and Caribbean	546,723,509	8.5 %	70,699,084	291.31 %	12.9 %	7.4 %
Oceania / Australia	33,443,448	0.5 %	17,655,737	131.7 %	52.8 %	1.8 %
<b>WORLD TOTAL</b>	<b>6,420,102,722</b>	<b>100.0 %</b>	<b>957,753,672</b>	<b>165.3 %</b>	<b>14.9 %</b>	<b>100.0 %</b>

NOTES: last updated in 2005, September, 30. The information regarding Internet usage is determined on the basis of publications of: [Nielsen/NetRatings](#), [International Telecommunications Union](#), local internet centers (NIC). Source: Miniwatts International, LLC

b) the usage of Internet in European countries group by EU members and non EU and the position of Romania

The usage of Internet in Europe						
EUROPE	People ( 2005 Estimated)	% percent from total population	Internet users	Grows 2000-2005	% People ( Penetration )	% usage at world level
European Union	460,270,935	7.2 %	225,006,820	141.5 %	48.9 %	23.5 %
Non EU Countries	270,747,588	4.2 %	48,256,135	385.8 %	17.8 %	5.0 %
<b>TOTAL EUROPE</b>	<b>731,018,523</b>	<b>11.4 %</b>	<b>273,262,955</b>	<b>165.0 %</b>	<b>37.4 %</b>	<b>28.5 %</b>
Other Countries	5,689,084,199	88.6 %	684,490,717	165.4 %	12.0 %	71.5 %
<b>WORLD TOTAL</b>	<b>6,420,102,722</b>	<b>100.0 %</b>	<b>957,753,672</b>	<b>165.3 %</b>	<b>14.9 %</b>	<b>100.0 %</b>
<b>ROMANIA</b>	<b>21,377,426</b>	<b>0.33%</b>	<b>4,940,000</b>	<b>517.5 %</b>	<b>23.1 %</b>	<b>1.8 %</b>

Source: Miniwatts International, LLC

The position of Romania in 2005 in the European area is described as:

<b>ROMÂNIA</b>	<b>% from total Population</b>	<b>% Internet usage</b>
<b>Reported to Europe</b>	2.92%	1.81%
<b>Reported to UE</b>	4.64%	2.20%
<b>Reported to non UE</b>	7.90%	10.24%

In 2012 the usage of Internet in Europe is characterized by the following figures (based on the sources as indicated in each table):

a) the Internet usage at world level by regions

<b>WORLD INTERNET USAGE AND POPULATION STATISTICS</b>						
<b>June 30, 2012</b>						
<b>World Regions</b>	<b>Population ( 2012 Est.)</b>	<b>Internet Users Dec. 31, 2000</b>	<b>Internet Users Latest Data</b>	<b>Penetration (% Population)</b>	<b>Growth 2000-2012</b>	<b>Users % of Table</b>
<a href="#">Africa</a>	1,073,380,925	4,514,400	167,335,676	15.6 %	3,606.7 %	7.0 %
<a href="#">Asia</a>	3,922,066,987	114,304,000	1,076,681,059	27.5 %	841.9 %	44.8 %
<a href="#">Europe</a>	820,918,446	105,096,093	518,512,109	63.2 %	393.4 %	21.5 %
<a href="#">Middle East</a>	223,608,203	3,284,800	90,000,455	40.2 %	2,639.9 %	3.7 %
<a href="#">North America</a>	348,280,154	108,096,800	273,785,413	78.6 %	153.3 %	11.4 %
<a href="#">Latin America / Caribbean</a>	593,688,638	18,068,919	254,915,745	42.9 %	1,310.8 %	10.6 %
<a href="#">Oceania / Australia</a>	35,903,569	7,620,480	24,287,919	67.6 %	218.7 %	1.0 %
<b>WORLD TOTAL</b>	<b>7,017,846,922</b>	<b>360,985,492</b>	<b>2,405,518,376</b>	<b>34.3 %</b>	<b>566.4 %</b>	<b>100.0 %</b>

NOTES: (1) Internet Usage and World Population Statistics are for June 30, 2012. (2) CLICK on each world region name for detailed regional usage information. (3) Demographic (Population) numbers are based on data from the [US Census Bureau](#) and local census agencies. (4) Internet usage information comes from data published by [Nielsen Online](#), by the [International Telecommunications Union](#), by [GfK](#), local ICT Regulators and other reliable sources. (5) For definitions, disclaimers, navigation help and methodology, please refer to the [Site Surfing Guide](#). (6) Information in this site may be cited, giving the due credit to [www.internetworldstats.com](#).

Copyright © 2001 - 2013, Miniwatts Marketing Group. All rights reserved worldwide.

b) the position of Romania in 2012 in the European area is, as percent of users from total population (that is bad estimated UP), on the last place (even we use the real number of the population as determined in 2012 the percent becomes 47,92% the place do not changes).

<b>Internet Users in the European Union</b>					
<b>EUROPEAN UNION</b>	<b>Population ( 2012 Est. )</b>	<b>Internet Users, 30-June-12</b>	<b>Penetration (% Population)</b>	<b>Users % Table</b>	<b>Facebook 31-Dec-12</b>
<a href="#">Austria</a>	8,219,743	6,559,355	79.8 %	1.8 %	2,915,240
<a href="#">Belgium</a>	10,438,353	8,489,901	81.3 %	2.3 %	4,922,260
<a href="#">Bulgaria</a>	7,037,935	3,589,347	51.0 %	1.0 %	2,522,120
<a href="#">Cyprus</a>	1,138,071	656,439	57.7 %	0.2 %	582,600
<a href="#">Czech Republic</a>	10,177,300	7,426,376	73.0 %	2.0 %	3,834,620
<a href="#">Denmark</a>	5,543,453	4,989,108	90.0 %	1.4 %	3,037,700
<a href="#">Estonia</a>	1,274,709	993,785	78.0 %	0.3 %	501,680
<a href="#">Finland</a>	5,262,930	4,703,480	89.4 %	1.3 %	2,287,960
<a href="#">France</a>	65,630,692	52,228,905	79.6 %	14.2 %	25,624,760
<a href="#">Germany</a>	81,305,856	67,483,860	83.0 %	18.3 %	25,332,440
<a href="#">Greece</a>	10,767,827	5,706,948	53.0 %	1.6 %	3,845,820

<a href="#">Hungary</a>	9,958,453	<b>6,516,627</b>	65.4 %	1.8 %	4,265,960
<a href="#">Ireland</a>	4,722,028	<b>3,627,462</b>	76.8 %	1.0 %	2,183,760
<a href="#">Italy</a>	61,261,254	<b>35,800,000</b>	58.4 %	9.7 %	23,202,640
<a href="#">Latvia</a>	2,191,580	<b>1,570,925</b>	71.7 %	0.4 %	414,520
<a href="#">Lithuania</a>	3,525,761	<b>2,293,508</b>	65.1 %	0.6 %	1,118,500
<a href="#">Luxembourg</a>	509,074	<b>462,697</b>	90.9 %	0.1 %	227,520
<a href="#">Malta</a>	409,836	<b>282,648</b>	69.0 %	0.1 %	217,040
<a href="#">Netherlands</a>	16,730,632	<b>15,549,787</b>	92.9 %	4.2 %	7,554,940
<a href="#">Poland</a>	38,415,284	<b>24,940,902</b>	64.9 %	6.8 %	9,863,380
<a href="#">Portugal</a>	10,781,459	<b>5,950,449</b>	55.2 %	1.6 %	4,663,060
<a href="#">Romania</a>	21,848,504	<b>9,642,383</b>	44.1 %	2.6 %	5,374,980
<a href="#">Slovakia</a>	5,483,088	<b>4,337,868</b>	79.1 %	1.2 %	2,032,200
<a href="#">Slovenia</a>	1,996,617	<b>1,440,066</b>	72.1 %	0.4 %	730,160
<a href="#">Spain</a>	47,042,984	<b>31,606,233</b>	67.2 %	8.6 %	17,590,500
<a href="#">Sweden</a>	9,103,788	<b>8,441,718</b>	92.7 %	2.3 %	4,950,160
<a href="#">United Kingdom</a>	63,047,162	<b>52,731,209</b>	83.6 %	14.3 %	32,950,400
<b>Total European Union</b>	<b>503,824,373</b>	<b>368,021,986</b>	<b>73.0 %</b>	<b>100.0 %</b>	<b>192,746,920</b>

NOTES: (1) The European Union Internet Statistics were updated for June 30, 2012. (2) Population is based mainly on data from the [US Census Bureau](#). (3) The Internet usage numbers come from various sources, mainly from data published by [Nielsen Online](#), [ITU](#), [GfK](#), Facebook and other trustworthy sources. (4) Data may be cited, giving due credit and establishing an active link to [Internet World Stats](#). Copyright © 2013, Miniwatts Marketing Group. All rights reserved worldwide.

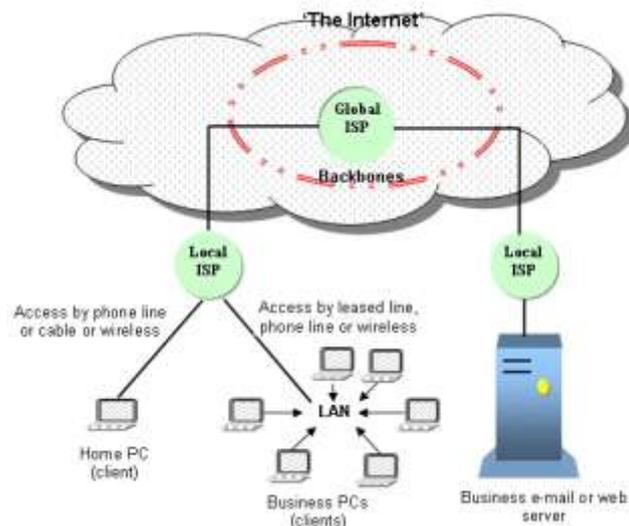
The Internet is built using hardware and software. Both hardware and software serves as a means to access what you really want: the information that the Internet can bring to your personal computer. Without the right hardware and software, you could not connect to the Internet, but having the hardware alone won't get you to the World Wide Web.

The Internet has two aspects, physical and logical and it can be viewed as a collection of physical and logical pieces that are tied together physically and logically:

- The **physical aspect** is a collection of wires, optical fibers, and microwave radio links and other devices that carry digital signals between computers. The combination of connections forms a redundant network. Computers are linked to one another in a web that provides multiple signal paths between any two machines. Today the term used for that “computers” here is **hosts** or **end systems** represented by a plethora of “computerized devices” such as TVs, laptops, gaming consoles, cell phones, Web cams, automobiles, environmental sensing devices, picture frames, and home electrical and security systems that are connected to Internet;
- The **logical aspect** is a set of standards for the signals that travel through that network. The Internet uses various protocols depending on what kind of data is being transferred. The languages that allow computers to talk to another are called protocols. The protocol is the method in which the network interface cards (NIC) communicate over the topology. Protocols are essentially electronic rules of behavior that allow the network interface cards to initiate and maintain communication. These rules are controlled by the protocol engine that:
  - accepts raw data from the sending source;
  - assembles and addresses packets;
  - attaches any necessary information such as internet routing;
  - places the packets onto the communication channel.

The Internet was not designed to link computers but to tie together computer networks and, consequently, to allow data to flow between networks. End systems are connected together by a network (in a final shape of a web) of communication links and packet switches. The end systems run protocols that control the sending and receiving of information within the Internet. The chief protocol and the defining standard of the Internet is TCP/IP (Transmission Control Protocol/Internet Protocol). Even if you only have a single personal computer when you connect with the Internet you must run a network protocol that allows your computer logically communicate to others. The common way to make an individual personal computer a physical part of Internet is to use a modem.

Internet allows communication between millions of connected computers world-wide. Information is transmitted from client PCs (individuals or companies) whose users request services to server computers (figure 1.2) that hold information and host business applications that deliver the services in response to request.



**Figure 1.2 Some infrastructure components of Internet**

The client PCs within homes or business are connected to Internet via local Internet Service Provider (ISP) which, in turn, are linked to larger ISPs with connection to the major national and international infrastructure or backbone (high-speed data transport channels).

End systems access Internet by intermediate of an Internet Service Provider that is itself a network of packet switches and communication links. The ISPs include residential ISPs (such as local cable or telephone companies), corporate ISPs, University ISPs, WiFi ISPs (such as those offering WiFi in the airports, coffee shops, malls, and other public places). ISPs provide Internet access also to content providers, connecting Web sites directly to the Internet. The ISPs are organized on several tiers between the lower-tier, where the ISPs are interconnected through national and the upper-tier, where the ISPs are interconnected through international high-speed fiber-optic links. Tier 1 ISPs are known as Internet backbone networks (Sprint, Verizon, MCI, AT&T, NTT, Level3, Quest and Cable&Wireless). The next tier is a client of the 1<sup>st</sup> tier Tier-1 and so on. Internet is also an infrastructure that provides services to applications such as electronic mail, Web navigation, instant messaging, Voice-over-IP (VoIP), Internet radio, video streaming, games/ distributed games, peer-to-peer file sharing, television over the Internet, remote access etc. The applications that requires multiple end systems to exchange data with each other are said distributed applications. The end systems attached to Internet provide an API (Application Programming Interface) that specifies, in terms of rules that software must follow, how software piece running in one end system asks the Internet infrastructure to deliver data to a specific destination software running on another end system. A client program is a program running on one end system that request and receives a service from a server program running on another and system. Since a client program typically runs on one computer and the server program runs on another computer, client-server Internet applications are, by definition, distributed applications [KR, 2010:12].

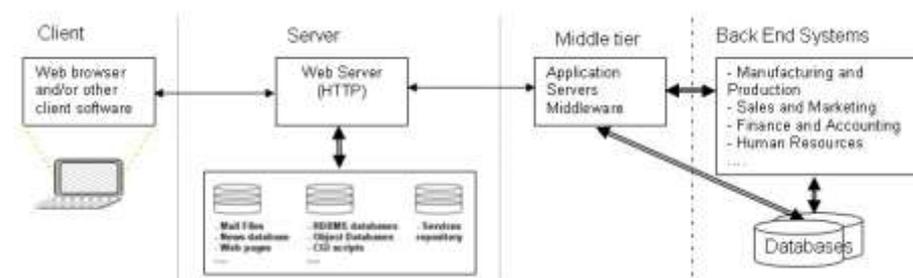
The **World Wide Web** (or web or www, for short) is a medium for publishing information on the Internet in an easy-to-use form. The medium is based on a standard document format known as HTML (Hypertext markup language). The www represented by all the interlinked documents on the Internet made up of pages containing text, graphics and other elements. The web is accessed using a web browser that enables user to navigate through the information available and display any page of interest

The transmission of information across the Internet is often described as being based around either pull or push technology:

- *Pull technology* describes information sent out as a result of receiving specific request, for example a page is delivered to a web browsers in response to a specific request from the user;
- *Push technology* describes information that is sent without a user's specifically requesting it, for example a customized news service received by subscribing to a channel or e-mail.

**Client/Server Technology.** The Internet is based on **client/server** technology (figure 1.3). All data, including e-mail messages and Web pages, are stored on server. The individuals access that resources and the net control through client applications, such as Web browser. A client uses the Internet to request information or services from a distant computer and the server sends the request information back to the client via Internet. The client platforms include a variety of devices and information appliance. An information appliance is a device (such as Internet-enabled cell phones, for example) that has been customized to perform, in a user friendly way, a few specialized computing tasks. In the following table are listed some common Internet platforms:

Device	Description
<b>PC</b>	General purpose computing platform that can perform many different tasks. The performed tasks can be complex to use
<b>Net PC</b>	Network computer with minimal local storage and processing capabilities and designed to use software and services delivered over the networks and the Internet
<b>MID</b>	Mobile Internet Device is a highly portable Internet-connected device both business and individual consumers designed as a pocket-size solution for access information on-the-go
<b>Smart Phone</b>	Provide voice communication and in addition has a small screen and keyboard for browsing the Web and exchanging e-mail
<b>Game machine</b>	Game machines provided with a modem, keyboard, and capabilities to function as Web access terminal
<b>PDA</b>	Wireless handheld personal digital assistant (PDA) with e-mail and Internet services. Typical functions for PDAs include address book, appointment scheduler, calculator, clock, expense tracking, currency conversions, alarm etc. Sophisticated PDA can include communications, spreadsheet and word processing applications
<b>E-mail machine</b>	Telnet with keyboard that provides textual e-mail capabilities (it requires linking to an e-mail service)
<b>Set-top box</b>	Is an important component of the Interactive digital TV system and is used to receive and decode message (from cable, satellite dish, aerial antenna etc) and then display on a conventional TV. It provide also surfing and e-mail capabilities using a television set and wireless keyboard (or remote control). The set-top box includes a modem that is used to pass back selections made on interactive channels (such as the interactive shopping channels, for example) .



**Figure 1.3 Client/server computing on the Internet**

In the right side of the figure we consider the **back end systems** (or back office) that are in use by enterprises. The enterprise software consists of a set of interdependent modules for applications such as sales and distribution, financial and accounting, investment management, production planning, plant maintenance and human resources etc that allows data to be used by multiple functions and business processes for more precise coordination and control. The modules can

communicate with each other directly or by sharing common repository data. Contemporary enterprise system uses client/server computing architecture.

In the companies in operation before PC and Internet appears we can find many existing legacy mainframe applications that are essential to daily operations and very risky to change. Numerous traditional and old in existence companies have departmental systems, which often-times did not share information and becomes in time “information silos” within the organization. These numerous disparate systems created the need for many distinct interfaces between those systems to make them communicate and cooperate. On another hand these legacy systems are incompatible with the new applications developed for PC platforms. In the same time a new imperative change imposed for business: the software used by these departments must be integrated and must be capable to adapt to the rapidly change in industries, technologies, environmental factors affecting businesses.

The legacy systems can be made more useful if their information and business logic can be integrated with other applications. One way to integrate various legacy systems is by using special software called middleware. **Middleware** is a special software which allows different software applications to communicate (it allows and assists data transfers between incompatible systems similarly to the way the network gateway operates in Internet).

Another way to integrate the existing systems is the use of an **enterprise application integration software (EAI)**. This kind of software is dedicated to tie together multiple applications to support enterprise integration. The software allows system builders to model their business process graphically and define the rules that applications should follow to make this process work. The software then generates the under-laying program instructions to link existing applications to each other to support those processes. To fulfill the new requirements of information systems, a new breed of software system called ERP (Enterprise Resource Planning) was created. ERP systems provide a single source of data with designed integration between different functional modules (such as Accounting, Sales and Distribution, Human Resources, Materials Management, Production Planning etc) to take full advantage of the enterprise’s stored information.

**Cloud technologies.** For Cloud computing we don’t have yet a definition unanimously accepted. Some of these definitions are:

- a) A massive network of servers or even individual PCs interconnected in a grid. The computers run in parallel, combining the resources of each to generate supercomputing-like power. [Google]
- b) A cloud is a pool of virtualized computer resources that hosts a variety of different workloads and allow them to be deployed and scaled-out through the rapid provisioning of virtual machines or physical machines; supports redundant, self-recovering, highly scalable programming models and resource usage monitoring in real time to enable rebalancing of allocation when needed.
- c) “Cloud computing is a style of computing where massively scalable IT-related capabilities are provided as a service across the Internet to multiple external customers [Gartner].”
- d) According to [LCT] “Cloud computing is a paradigm that focuses on sharing data and computations over a scalable network of nodes.” The computing cloud is a massive network of nodes having at least a two dimensional scalability:

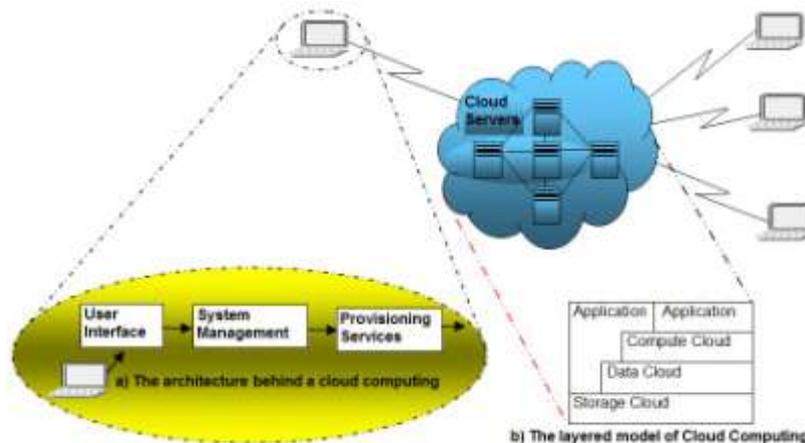
- *horizontal* - as the ability to connect and integrate multiple clouds to work as a single logical cloud;
- *vertical* - as the ability to improve the capacity of a cloud by enhancing individual existing nodes in the cloud.

According to [IBM-09] Cloud computing is:

- a business delivery model by which hardware, software and network resources are optimally leveraged to provide innovative services over the Web, and servers are provisioned in accordance with the logical needs of the service using advanced, automated tools. The business model of a cloud facilitates more efficient use of existing resources.
- an infrastructure management methodology that enables IT organizations to manage large numbers of highly virtualized resources as a single large resource. It also allows IT organizations to massively increase their data center resources without significantly increasing the number of people traditionally required to maintain that increase.

The cloud enables the service creators, program administrators and others to use these services via a Web-based interface that abstracts away the complexity of the underlying dynamic infrastructure. The cloud also provides a user interface that allows both the user and the IT administrator to easily manage the provisioned resources through the life cycle of the service request. The cloud user disposes of self-service functions (that can be performed 24 hours a day and take only minutes to perform) to add/remove servers, change the installed software, increase/decrease the allocated processing power, memory or storage and even can start, stop and restart servers.

Figure 1.4 gives a image about how the cloud build and how user realizes the connection to cloud computing together with a closer look to the user (what is before connection) and to a layered approach of the cloud. The layers in the architecture are defined as categories of services:



**Figure 1.4 Connecting to cloud**

- Storage Cloud - storage services;
- Data Cloud - data management services (record, column, or object-based);
- Compute Cloud - computational services;
- Application - generally SaaS.

The user access the cloud, for the services provisioned by the vendor from a browser application program running anywhere in the world, by intermediate of his user interface and by using the services of the system management.

The cloud computing is a logical corollary and consequence of many ancestors: grid computing, utility computing and Software-as-a-Service, as shown in figure 1.5.

			➔
Grid Computing	Utility Computing	Software-as-a-Service	Cloud Computing
- Solving large problems with parallel computing - Made mainstream by Globus Alliance (1980)	- Offering computing resources as a metered service - Introduced in late of 1990	- Network-based subscriptions to applications - Gained momentum in 2001	- Next-generation Internet computing - Next-generation data-centers

<p><i>Usually a grid is a cluster of servers on which a large task could be divided into smaller tasks to run in parallel. The applications must conform to the grid software interfaces.</i></p>	<p><i>On-demand computing</i></p>		<p><i>Computing and extended IT and business resources, such as servers, storage, network, applications and processes, can be dynamically shaped or carved out from the underlying hardware infrastructure and made available to a workload.</i></p>
<p><b>Figure 1.5 Connecting to cloud (Adapted from IBM-09)</b></p>			

The cloud is easy to program than distributed or grid computing.

The clouds can be specialized such as cloud storage, cloud services, calculation cloud, etc.

Related to cloud computing we have the following concepts:

- *Cloud storage* - data are stored on a virtual server having a dynamically location perceived by the user as a static one.
- *Cloud services* - any web application or service offered via “cloud computing” is called “Cloud service”. The user runs the application stored in cloud by intermediate of his web browser. If the user computer fails this fact do not affect both application and data. By storing the documents in the cloud is possible that all users granted to access and manipulate the document to work simultaneously as a team on this.
- *Software-as-a-Service (SaaS)* - a sole application is delivered to thousand of users by intermediate of vendor servers. Each organization deserved by vendor is called tenant, and the architecture of this arrangement is called multi-tenant architecture. The clients do not pay for the software possession, as in a desktop licensed usage, instead they pay for usage based on a time scale and a subscription. The vendor servers are partitioned virtually so that each deserved organization works with an instance of the application virtually personalized (customized). The most known application offered by Cloud computing is Google MapReduce, that run on a cloud composed by 1,800 machines 2 GHz Intel Xenon, 4GB memory and 160 GB IDE disks. The estimation of Gartner is that SaaS will rise early at a rate of 22.1% until 2011.

From a Google point of view (one of the bigger supplier in Cloud resources) the Cloud computing is:

- *user-centric* - once connected a user can access the stored objects and share with others and any device accessing his data becomes as if is his object;
- *task-centric* - is focused on application result and not on the application itself;
- *powerful* - thousands of computers connected together;
- *accessible* - any computer having a connection to Internet (for efficiency considerations, a broadband connection) can use the cloud;
- *intelligent*;
- *programmable*.

The Cloud computing represents for giant IT companies a strategic field of investments in hardware, software and research:

- IBM and Dell ship cloud computing machines.
- Google have in 2008 1 million servers in 30 data-centers and realizes early investments of about \$2 Billions in Datacenters.
- A new IBM-Google initiative aims to provide computer science students with a complete suite of open source-based development tools so they can gain the advanced programming

skills necessary to innovate and address the challenges of this computing model which uses many computers networked together through open standards and thereby drive the Internet's next phase of growth [IBM].

- Microsoft enlarge their server farms at a rate of 20,000 new servers/month (2008).

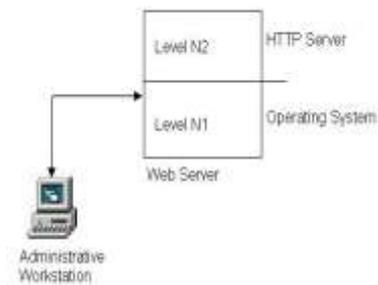
### 1.1.1 The Logical Structure of Web Servers

The base plate of a web server (figure 1.6) composed by three basic elements: the physical server, the server operating system (must include a network operating system - NOS) and the server called HTTP (HyperText Transport Protocol).

The physical structure, processor, network cards, connection and operating system are described in the books indicated by the references [AvDg03 and DgAv05] and will not be reintroduced here.

Very briefly, in the functional architecture from figure 1.6, the elements are:

- Administrative workstation (or the system console) is the workstation used to administer the operating system running on the server. In general, a Web server, that is part of Internet, is a dedicated server (it is possible to use non-dedicated servers in Intranet configuration, this means at local level and, in these situations the machine running the server operating system can be also used as a workstation and consequently can play the role of the administrative workstation).
- Servers uses a Network Operating System (NOS) that must enough capable to offer simultaneously services to most clients. The server, utilizing NOS acts the same as a network traffic police which controls the Workstation file requests (reads and writes to network drives), printer output and communications between users and file servers attached to the network. This is the system software necessary to control the access to and flow of information around the network. It is used to implement the different levels of the open system interconnection (OSI) model. It provides the following functions:
  - access control or security through providing user accounts with user names and passwords;
  - file and data sharing of data stored on a database server or file server;
  - communication between users via e-mail, diary systems or workgroup software;
  - sharing of devices.



**Figure 1.6 The functional architecture of the base plate of a web server**

The Operating System of the Network can be UNIX (and anyone of his clones such as Linux), MacOS, OS/2, Novell NetWare, IBM LAN Manager (these last previous two are the most widely used), Banyan Vines, Windows NT xx Server, Windows 2000 or 2003 Server, etc;

- HTTP is the protocol that governs how web browsers (clients) and web servers talk to each other. All messages sent between browsers and servers must be formatted according to the HTTP specification. The HTTP commands allow an application to interpret a page together with his HTML (HyperText Markup Language) links. The HTTP server manages, interprets and acts the HTTP commands.

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World-Wide Web global information initiative since 1990. The first version of HTTP, referred to as HTTP/0.9, was a simple protocol for raw data transfer across the Internet. HTTP/1.0, as defined by RFC1945, improved the protocol by allowing messages to be in the format of MIME like messages, containing meta-information about the data transferred and modifiers on the request/response semantics. However, HTTP/1.0 does not sufficiently take into consideration the effects of hierarchical proxies, caching, the need for persistent connections, or virtual hosts. In addition, the

proliferation of incompletely implemented applications calling themselves “HTTP/1.0” has necessitated a protocol version change in order for two communicating applications to determine each other’s true capabilities. This specification defines the protocol referred to as “HTTP/1.1”. This protocol includes more stringent requirements than HTTP/1.0 in order to ensure reliable implementation of its features. Practical information systems require more functionality than simple retrieval, including search, front-end update, and annotation.

The system software manufacturers offer a lot of web servers that runs under different platforms. Table 1.1 shows the main pairs server-platform on the market.

Table 1.1 The main pairs server-platform

Manufacturer	Server	Platform
Apache	Apache <sup>*)</sup>	Linux, Unix, Windows (NT, 2k, 2003, 2008, ...)
Netscape	Enterprise server	Linux, Unix, Windows (NT, 2k, 2003, ...)
Microsoft	Internet Information Services	Windows (NT, 2k, 2003)
Lotus	Domino	Windows NT, OS/2
Novell	Intranetware	Netware, Windows (2k, 2003)
Sun	Sunserver	Solaris
Oracle	Webstart	Unix

<sup>\*)</sup> is one of the pairs widely used in the domain of web servers

In the table below is shown the position of web server top developers:

Developer	January 2006	Percent	February 2006	Percent	Change
Apache	50,502,840	67.11	51,810,676	68.01	0.90
Microsoft	15,510,953	20.61	15,666,702	20.56	-0.05
Sun	1,879,856	2.50	1,880,313	2.47	-0.03
Zeus	561,524	0.75	579,198	0.76	0.01

Source: Web Server Survey news.Netcraft.com February 2006 survey based on received responses from 76,184,000 sites

Developer	July 2009	Percent	August 2009	Percent	Change
Apache	113,019,868	47.17%	104,611,555	46.30%	-0.87
Microsoft	55,918,254	23.34%	49,579,507	21.94%	-1.39
qq.com	30,447,369	12.71%	30,278,988	13.40%	0.69
Google	14,226,904	5.94%	14,213,976	6.29%	0.35
nginx	10,174,573	4.25%	11,502,109	5.09%	0.84
lighttpd	1,326,240	0.55%	2,025,521	0.90%	0.34

Source: Web Server Survey news.Netcraft.com

The HTTP protocol is a request/response protocol. The HTTP protocol allow to clients and web servers to establish a connection based on TCP (Transmission Control Protocol) allowing data transfers (documents, images etc) from server to client or from client to server. A client sends a request to the server in the form of a request method, URI (Uniform Resource Identifiers), and protocol version, followed by a MIME-like (Multipurpose Internet Mail Extensions) message containing request modifiers, client information, and possible body content over a connection with a server. The server responds with a status line, including the message’s protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta-information, and possible entity-body content. URI’s have been known by many names: WWW addresses, Universal Document Identifiers, Universal Resource Identifiers, and finally the combination of Uniform Resource Locators (URL) and Names (URN). As far as HTTP is concerned, Uniform Resource Identifiers are simply formatted strings which identify - via name, location, or any other characteristic - a resource.

The data transferred have an associated data-type (a header describing the content text, image, HTML etc and how they coded) and the transfer uses the ASCII character set and the MIME standard.

The information using the MIME standard are converted as MIME standard requires and provided with a header having, for example, the following fields:

*MIME-version: 1.0*  
*Content-Type: type/specification parameter\_name=parameter\_value*  
*Content-ID:*  
*Content-Description:*

In the Content-Type field is specified the type of the send message body. The specifications of document types in MIME standard are listed in the table 1.2.

**Table 1.2 The MIME standard document types**

<b>Specification</b>	<b>Explanation</b>
Text/plain	Unformatted text
Text/richtext	Text with simple formatting elements
Text/enriched	Text with complex formatting elements
Text/html	Text with HTML formatting elements
Image/jpeg	Image in JPEG format
Image/gif	Image in GIF format
Audio/basic	ISDN format on 8 bits, 1 channel and 8000 Hz
Video/mpeg	Movie in MPEG format
Message/external-body	Reference to an unformatted document stored on hard drive
Message/rfc822	Document in RFC822 (e-mail) format
Message/partial	The biggest part of a document in RFC822 (e-mail) format
Multipart/mixed	The content composed by many documents in MIME format
Multipart/alternative	The contents composed by many parts in MIME format each part containing the same information but represented in a different format
Multipart/parallel	The contents includes many parts in MIME format that can be processed simultaneously
Multipart/digest	The MIME body have many parts each of each in message/rfc822 format
Application/octet-stream	Can not be processed by the program and requires saving the MIME body in a
Application/postscript	Document or application in PostScript format
Application/x-www-form-urlencoded	Data from HTML forms

The field Content Transfer Encoding describes the method used for data coding in the MIME body as shown in table 1.3.

**Table 1.3 The types for content data coding**

<b>Type</b>	<b>Explanation</b>
7bit	The contents is in NVT ASCII format, un-coded
8bit	The content composed by rows containing characters represented on 8 bits, uncoded
Binary	The content composed by different characters but not divided in rows

Quoted-printable	The contents is coded in NVT ASCII format on 7 bits using the q method
Base64	The contents is coded using b method (base64)
x-user	The contents coded with a user defined method

The client application (a general browser or another web oriented application) contact the *http* server and then send his request in which it specifies the type of action the browser wishes the server to perform. The server applications, executes the client request and, send to this one an answer including the information corresponding to the query execution. The typical structure of a client query is: *method identifier, required object name, the client http protocol version number*. In the context of HTTP, a method is essentially the name of a command. The HTTP methods identifiers and the action requested are listed in table 1.4.

[Internet Society, RFC 2616] HTTP messages consist of requests from client to server and responses from server to client:

**HTTP-message = Request | Response ; HTTP/1.1 messages**

Request and Response messages use the generic message format of RFC 822 for transferring entities (the payload of the message). Both types of message consist of a start-line, zero or more header fields (also known as “headers”), an empty line (i.e., a line with nothing preceding the CRLF) indicating the end of the header fields, and possibly a message-body.

**generic-message = start-line  
\*(message-header CRLF)  
CRLF  
[ message-body ]**

**start-line = Request-Line | Status-Line**

The request-header fields allow the client to pass additional information about the request, and about the client itself, to the server. These fields act as request modifiers, with semantics equivalent to the parameters on a programming language method invocation (Some of this are Accept, Accept-Charset, Authorization, Host, User-Agent etc; see the RFC 2616 for any details). After receiving and interpreting a request message, a server responds with an HTTP response message:

**Response = Status-Line  
\*(( general-header  
| response-header  
| entity-header ) CRLF)  
CRLF  
[ message-body ]**

The first line of a Response message is the Status-Line, consisting of the protocol version followed by a numeric status code and its associated textual phrase, with each element separated by SP characters. No CR or LF is allowed except in the final CRLF sequence.

**Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF**

The Status-Code element is a 3-digit integer result code of the attempt to understand and satisfy the request. The first digit of the Status-Code defines the class of response:

- 1xx: Informational - Request received, continuing process
- 2xx: Success - The action was successfully received, understood, and accepted
- 3xx: Redirection - Further action must be taken in order to complete the request
- 4xx: Client Error - The request contains bad syntax or cannot be fulfilled
- 5xx: Server Error - The server failed to fulfill an apparently valid request

**Table 1.4 Method Identifiers in HTTP protocol**

<b>Method</b>	<b>Explanation</b> (The hachured text in the Explanation column refers to HTTP/1.1 specification as defined in RFC 2616 Internet Society, June 1999)
GET	The client want download the content of the object specified in the query. The GET method means to retrieve whatever information

	(in the form of an entity) is identified by the Request-URI. If the Request-URI refers to a data-producing process, it is the produced data which shall be returned as the entity in the response and not the source text of the process, unless that text happens to be the output of the process.
HEAD	The client wants in advance the http header fields as if he receives from server when download completed. The HEAD method is identical to GET except that the server must not return a message-body in the response. The metainformation contained in the HTTP headers in response to a HEAD request should be identical to the information sent in response to a GET request. This method can be used for obtaining metainformation about the entity implied by the request without transferring the entity-body itself. This method is often used for testing hypertext links for validity, accessibility, and recent modification.
POST	The client wants modify the required object by sending in query the changed contents of the object. The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. POST is designed to allow a uniform method to cover the following functions: <ul style="list-style-type: none"> <li>○ Annotation of existing resources;</li> <li>○ Posting a message to a bulletin board, newsgroup, mailing list, or similar group of articles;</li> <li>○ Providing a block of data, such as the result of submitting a form, to a data-handling process;</li> <li>○ Extending a database through an append operation.</li> </ul>
PUT	The client want store transmitted data to the URL address specified in the query. The PUT method requests that the enclosed entity be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource, the enclosed entity should be considered as a modified version of the one residing on the origin server. If the Request-URI does not point to an existing resource, and that URI is capable of being defined as a new resource by the requesting user agent, the origin server can create the resource with that URI. If a new resource is created, the origin server must inform the user agent via the 201 (Created) response (status-code value 201). If an existing resource is modified, either the 200 (OK) or 204 (No Content) response codes should be sent to indicate successful completion of the request. If the resource could not be created or modified with the Request-URI, an appropriate error response should be given that reflects the nature of the problem. The recipient of the entity must not ignore any Content-* (e.g. Content-Range) headers that it does not understand or implement and must return a 501 (Not Implemented) response in such cases.
PATCH	Similarly to PUT bat the body contains only the changes to be done on the object given by URL
COPY	The clients wants copy the resource specified by URL
MOVE	The client wants change the name of the resource given by URL

DELETE	The client wants delete the resource specified by URL. The DELETE method requests that the origin server delete the resource identified by the Request-URI. This method may be overridden by human intervention (or other means) on the origin server. The client cannot be guaranteed that the operation has been carried out, even if the status code returned from the origin server indicates that the action has been completed successfully. However, the server should not indicate success unless, at the time the response is given, it intends to delete the resource or move it to an inaccessible location.
LINK	The client want link the resources specified in the query
UNLINK	The client want unlink the resources specified in the query
TRACE	The client wants that the server include in the answer what this receive from his part. The TRACE method is used to invoke a remote, application-layer loop-back of the request message. The final recipient of the request should reflect the message received back to the client as the entity-body of a 200 (OK) response. The final recipient is either the origin server or the first proxy or gateway to receive a Max-Forwards value of zero (0) in the request. A TRACE request must not include an entity.
OPTIONS	The client wants supplemental information about the features offered by the specified resource. The OPTIONS method represents a request for information about the communication options available on the request/response chain identified by the Request-URI. This method allows the client to determine the options and/or requirements associated with a resource, or the capabilities of a server, without implying a resource action or initiating a resource retrieval.
WRAPPED	Allows unifying sub-queries in one query.
CONNECT	This specification reserves the method name CONNECT for use with a proxy that can dynamically switch to being a tunnel (e.g. SSL tunneling).

\*) The hachured text in the Explanation column refers to HTTP/1.1 specification as defined in RFC 2616 Internet Society, June 1999

The structure of the server answer is similarly with the structure of the query and contains:

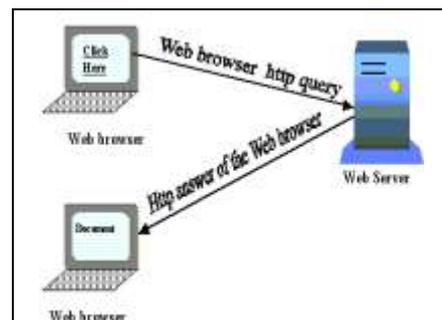
- the protocol version that must be used to process the server answer (the result of the execution of the operation specified by the client);
- the header fields;
- the http body.

The best view of the Internet comes with following a packet from your personal computer:

when you log into a web site, you actually send a command to a distant server telling it to download a page of data to your personal computer (figure 1.7).

Your web browser packages that command into a packet labeled with the address of the server storing the page that you want. Your personal computer sends the packet to your modem (or terminal adapter), which transmits it across your telephone or other connection to your Internet Service Provider (ISP).

The ISP actually operates as a message forwarder. At



**Figure 1.7 The principle of communication between a Web**

the ISP, your message gets combined with those from other PCs and sent through a higher speed connection (at least you should hope it is a high speed connection) to yet another concentrator that eventually sends your packet to one regional center. There the major Internet carriers exchange signals, routing the packets from your modem to the carrier that haul them to their destination based on their Internet address.

The World Wide Web is the most visually complicated and compelling aspect of the Internet. Despite its appearances, however, the web is nothing more than another file transfer protocol. When you call up a page from the web, the remote server simply downloads a file to your personal computer. Your web browser then decodes the page, executing commands embedded in it to alter the typeface and to display images at the appropriate place. Most browsers cache several file pages (or even megabytes of them) so that when you step back, you need not wait for the same page to download once again.

### 1.1.2 The “transport” protocols

TCP/IP groups together the communication protocols used to manage the data transmission over Internet. The description of TCP/IP was introduced in §1.8.

### 1.1.3 The IP addressing

Currently there are two types of Internet Protocol (IP) addresses in active use: IP version 4 (IPv4) and IP version 6 (IPv6). IPv4 was initially deployed on 1 January 1983 and is still the most commonly used version. IPv4 addresses are 32-bit numbers often expressed as 4 octets in "dotted decimal" notation (for example, 192.0.32.67). Deployment of the IPv6 protocol began in 1999. IPv6 addresses are 128-bit numbers and are conventionally expressed using hexadecimal strings (for example, 1080:0:0:0:8:800:200C:417A). The computers in TCP/IP based networks, even having only one computer, are called hosts. This name comes from the first deployment of TCP/IP – in the moment the standard defined the personal computers and workstation don't exist yet – all existing computers are multi-user and for that reason they called host. In this paragraph we explain the structure and usage of IP addresses in IP version 4.

In Internet each station has a unique number expressed as a 32-bit number and all of the Internet addresses are global. From the address itself, neither you nor a computer can tell where that address is or, more importantly, how to connect to it. The routers in the Internet regional centers maintain tables to help quickly send packets to the proper address. Without such guidance, packets wander throughout the world looking for the right address.

The Internet addresses are coded on 4 bytes and are expressed in so called **dotted-decimal** notation: for example for the number 2188611658 the address can be written (in 256 base) as  $130 \times 256^3 + 115 \times 256^2 + 144 \times 256^1 + 69 \times 256^0 \Rightarrow 130.115.144.69$

The Internet addresses are organized in five classes from A to E. Each address belonging in the class A, B or C consists of two parts:

**a)** a network identifier (**netid** – network address; we denote this by letter N), distributed by the non-governmental organization InterNIC (Internet Network Information Center - [www.internic.org](http://www.internic.org); [www.internic.net](http://www.internic.net)) or one of the regional centers. This address is used for routing over Internet (the choosing of the pathway from router to router);

**b)** a host identifier (**hostid** – the address of the machine in the network; we denote this by letter H). This address part can be also divided into two parts – sub-network address and the hardware address. The subnet address allows routing inside the private network that can provide routers or other interconnection devices that splits the network.

Both IPv4 and IPv6 addresses are assigned in a delegated manner. Users are assigned IP addresses by Internet service providers (ISPs). ISPs obtain allocations of IP addresses from a local Internet registry (LIR) or national Internet registry (NIR), or from their appropriate Regional Internet Registry (RIR):

[AfriNIC \(African Network Information Centre\)](#) - Africa Region

[APNIC \(Asia Pacific Network Information Centre\)](#) - Asia/Pacific Region

[ARIN \(American Registry for Internet Numbers\)](#) - North America Region  
[LACNIC \(Regional Latin-American and Caribbean IP Address Registry\)](#) – Latin America and some Caribbean Islands

[RIPE NCC \(Réseaux IP Européens\)](#) - Europe, the Middle East, and Central Asia

The Internet Assigned Numbers Authority - IANA - has the role to allocate IP addresses from the pools of unallocated addresses to the RIRs according to their established needs. When an RIR requires more IP addresses for allocation or assignment within its region, the IANA makes an additional allocation to the RIR.

The class D addresses are reserved for broadcast groups (multicast addressing) and E for future use (experimental).

The range values for the first classes (A, B, and C) classes are:

Class	The maximal number of networks	The maximal number of hosts per network	Address structure*	Comments
<b>A</b>	128	16777216	N.H.H.H	Major networks
<b>B</b>	16384	65536	N.N.H.H	Large sites
<b>C</b>	2097152	256	N.N.N.H	Small sites, or groups of midsize

\* N stand for network and H stand for host

The address range values for the five classes are:

Class	Address range		1 <sup>st</sup> byte
<b>A</b>	0.0.0.0	121.255.255.255	1-127
<b>B</b>	128.0.0.0	191.255.255.255	128-191
<b>C</b>	192.0.0.0	223.255.255.255	192-223
<b>D</b>	224.0.0.0	239.255.255.255	224-239
<b>E</b>	240.0.0.0	241.255.255.255	240-254

IP partitions the routing problem into three parts:

- routing exchanges between end systems and routers (ARP),
- routing exchanges between routers in the same routing domain (interior routing), and,
- routing among routing domains (exterior routing).

The machines having standard IP addresses can change information over Internet. Two machines that are members of two different networks must pass through an interconnection system of the network, a router (it is possible to pass through many routers to go from one machine to another one). Each router is connected at least to two machines.

The distinction between the hardware address and network address is realized by intermediate of a so called subnet mask. The comparison between the destination address of an IP packet and the subnet-mask shows if the receiver (the destination) is a member of the same network or not. If the receiver is not a member of the same network with the sender the packet is transmitted (passed) to router that decides, according with his routing table and other reasons (the traffic, for example), to which other router will be send. If the receiver is in the same network an address resolution frame of the logic address with the hardware (physical) address is send over the network. The receiver that recognize that address resolution frame (ARP) respond by giving his hardware address and from that moment the communication between the two machines can really take place.

It is still possible for almost people to get assignment of a number for a small "Class C" network in which the first three bytes identify the network and the last byte identifies the individual computer. Larger organizations can get a "Class B" network where the first two bytes identify the network and the last two bytes identify each of up to 64 thousand individual workstations. There are only about 2 million class A, B and C addresses. Almost all the "B" class addresses are assigned. As a result there is a proposal to enlarge the address space to 128 bits, called IPNG (Internet Protocol Next Generation or IPv6). It also removes certain non-essential features of the IP protocol making it faster and easier to implement.

Certain addresses have special meanings. In particular 0, 127 and 255 are usually reserved for special use. The number 255 indicates a broadcast address (for example

131.123.2.255), which is listened for by all machines on the net or subnet. Note that some vendors use 0 as the broadcast address by default (e.g. Sun) whereas others use 255. All systems on a network must be configured to use the same broadcast address. This is set with the operating system specific command. The value 0 is not assigned to any machine or network. The network address with 127 as the first byte is the "loopback network", which is fictitious. The address 127.0.0.1 is called "localhost" and means the current host machine.

The organization connects to the Internet through one of a dozen regional or specialized network suppliers. The network vendor is given the subscriber network number and adds it to the routing configuration in its own machines and those of the other major network suppliers.

When designing networks we generally build a network of networks using some devices, such as routers and bridges, allowing us to extend a network beyond the limits imposed by the standard on a single network. It turns out that routers or occasionally hosts acting as gateways play a special part in this design. Routers and gateways understand different protocols, such as IP, and can look at the IP portion of a packet and from the destination address to determine the route it should take next. The IP uses a concept called subnets to determine individual networks. Each separate network is a separate subnet, the router needs to look at IP addresses and determine if they belong to that network or not. This is done by using a subnet mask. The subnet mask is a 32 bit value as IP address is and is logically and-ed with the IP address to see if the destination is on the same network as the router or gateway. For a class B address a normal subnet mask value would be 255.255.255.0 or sometimes displayed with the hexadecimal address of 0xfffff00. Those values produce the equivalent mask, one with the first 24 bits set to 1 and then the remaining bits are 0.

Lets use an example to demonstrate how this works. Grivita building has an IP address of 130.85.105.3 and a subnet mask of 255.255.255.0. The router interface supporting that building has an IP address of 130.85.105.1 and a subnet mask of 255.255.255.0. When the router interface sees a packet with Grivita destination address it performs the logical and comparison on both its own IP address and subnet mask as well as the destination addresses IP address. It then compares the two resulting values, if they are equal the router knows the packet is on the same network and does not need to be forwarded. In the examples that follows are shown the both cases. In the sample the addresses expressed as dotted decimal addresses are translated into the equivalent of that in binary to easy apply the bitwise logical And operation (anded):

a) the result of applying the bitwise And operation produces the same value

Router address	Grivita address
130.85.105.1	130.85.105.3
10000010.01010101.01101001.00000001	10000010.01010101.01101001.00000011
<b>Anded with</b>	
subnet mask	subnet mask
255.255.255.0	255.255.255.0
11111111.11111111. 11111111.00000000	11111111.11111111. 11111111.00000000
<b>Result</b>	
10000010.01010101.01101001.00000000	10000010.01010101.01101001.00000000
130.85.105.0	130.85.105.0

b) the result of applying the bitwise And operation do not produces the same value<sup>\*)</sup>

Router address	Polytechnic University address
130.85.105.1	130.85.60.8
10000010.01010101.01101001.00000001	10000010.01010101.00111100.00001000
<b>Anded with</b>	
subnet mask	subnet mask
255.255.255.0	255.255.255.0
11111111.11111111. 11111111.00000000	11111111.11111111. 11111111.00000000
<b>Result</b>	
10000010.01010101.01101001.00000000	10000010.01010101. 00111100.00000000
130.85.105.0	130.85.060.0

<sup>\*)</sup> The results are not equal and the router must consult it's routing table to forward the packet on to the next destination.

## 1.1.4 The DNS

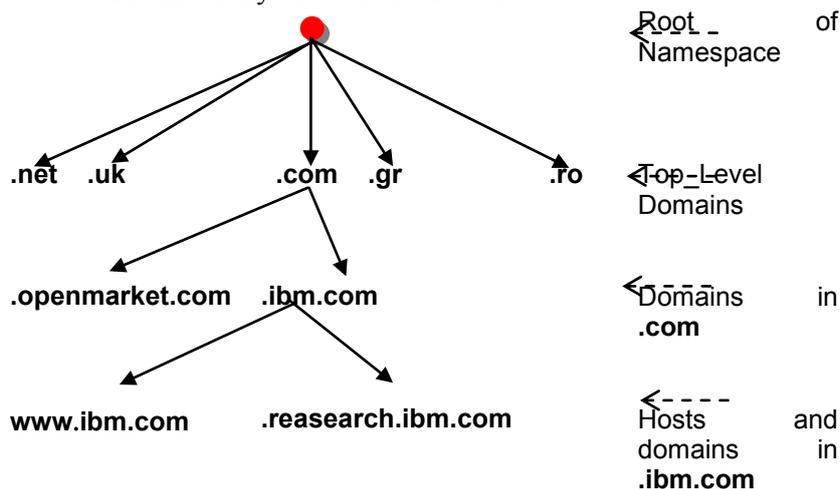
The usage of these dotted-decimal addresses can be very restrictive for common users. In Internet the common users access the servers and other shared resources by using **names**, almost of the time meaningful names. These names are associated to the dotted-decimal address of the station and are allowed and managed by DNS (Domain Name System).

DNS, was specified in 1983 and, allows the mapping of symbolic names to Internet addresses. Originally was realized statically in a centralized file (in Linux can be done statically in `/etc/hosts`). As Internet grew there was a need for a dynamic distributed system.

DNS defines:

- A hierarchical namespace for hosts;
- A host table implemented as a distributed database;
- Library routines for access ;
- Routing for e-mail;
- A protocol to exchange naming information.

The hierarchy structure for DNS is:



The DNS:

- Is organized as tree of domains with ascending authority;
- Offers two types of top-level domains
  - 3 letter in US (such as com, edu, gov, mil, net, org, intr, arpa etc). The “three” letters (can be more than three) can be used worldwide;
  - 2 letter national (ISO – such as ro, fr, uk, gr, us etc);
- Contains second level domains assigned by RIR (Regional Internet Registry), [InterNIC](#) (Internet Network Information Center) or RIPE in Europe, for example;
- Allows creating as desired, by organizations having second level domains, lower level sub-domains (e.g. ie.ase.ro)
- Allows delegating authority to create further sub-domains (e.g. vb.ie.ase.ro).

Note that domains reflect organizational structure whereas IP addresses reflect network connectivity (for routing purposes). These are often the same but do not need to be.

The letters defining a domain are used as Internet Domain Name Suffixes. The list of common Internet Domain Name Suffixes is shown in table 1.6.

Table 1.6 **Common Internet Domain Name Suffixes** (<http://www.icann.org>)

Ending	Kind of application
.arts	Cultural groups
.arpa	ARPAnet site (USA)
.aero	Air-transport industry
.asia	Asian Countries
.biz	Restricted to Business
.com	General business and individuals
.coop	Cooperatives
.edu or .ac	Schools/Educational sites
.eu	European Countries

.firm	Businesses
.gov	Government
.info	Information services (unrestricted use)
.int	International Institutions
.jobs	Human resource management
.mil	Military (USA)
.mobi	Mobile
.museum	Museums
.net	Internet service providers or general/administrative network
.nom	Individuals
.name	Individuals
.nato	NATO site
.org	Organizations
.pro	Accountants, lawyers, and physicians
.rec	Recreation sites
.store	Retailers
.travel	Travel related business
.web	Web-related organizations
.ro, .fr, .deu, .uk ...	The country domain

There are a number of "root nameservers" in existence in various corners of the Internet which store the ultimate information for the root domain, as well as zones for a handful of top-level domains. Certain organizational units, such as countries and universities, have delegation of domains underneath the root and top-level domains. Entities wishing domain names must register, and perhaps receive delegation of, their domains from the appropriate registry.

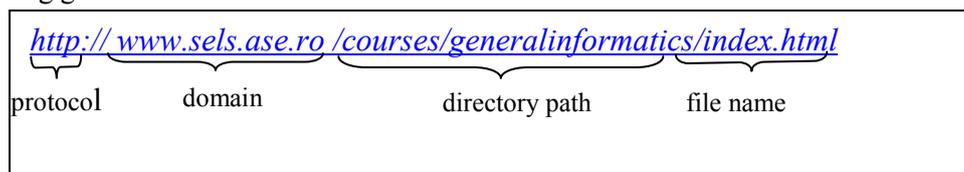
### 1.1.5 URL

Web pages and related files are located and accessed in Internet by means of special constructions called URL. Internet addresses are separate and distinct from the domain names used as **Uniform Resource Locators (URLs)** through which you specify Web pages. The domain names give you a handle with a natural-language look. Internet addresses are, like everything in computing, binary codes. Even domain names are running short. Finding a clever and meaningful name for a web site is a challenge that's ever increasing. Believing that one of the problems in the shortage of URLs has been the relatively few suffixes available, one of the coordinating agencies for Internet names, the International Ad Hoc Committee, proposed seven additional suffixes in addition to the six already in use in the U.S. and the national suffixes used around the world (the ISO country two letters such as .ro for Romania, .us for United States, .uk for United Kingdom, .fr for France, and so on).

URL is an acronym for Uniform Resource Locator. URL is expressed as a character string that supplies the Internet address of a site or of a www resource. The general syntax of URL is:

*communicationservice://hostname[:portnumber]/pathname/resourcename*

In a Web page the links are represented by specially formatted text strings or by graphical elements that, when acted (by a mouse click, for example) displays more text or graphics. This files tagged by links can be represented by other Web pages or any kind of files such as graphic, image, sound, video, data fill-in forms, Java applets, movies and any kind of necessary file. A hypothetical URL can take the following general structure:



An URL address can consist of five parts: protocol, domain, directory path, file name and anchor. For these elements a brief description follows:

- **protocol:** represented by rules that governs the data transfer in the network. Internet uses for Web pages (HTML pages) the http (HyperText Transport Protocol) – the word in the example URL *http:*
- **domain:** represented by the name of the host computer (hostname) and the Internet namespace - [www.sels.ase.ro](http://www.sels.ase.ro)
- **directory path:** the absolute or relative location of the file - [courses/generalinformatics](http://www.sels.ase.ro/courses/generalinformatics)

- **file name:** the web page, graphic, or sound file - *index.html*
- **anchor:** a marker which identifies a location inside a file (like a bookmark in normal documents) to which you can link. Once an anchor is placed in a location you can create a link to that spot.

The most common URL type is:

- *file://* - a local URL located in your hard drive(s) for example <file://c:/index.htm> that points the file called index.htm stored in the root of the local drive C: (Windows, MS-DOS);
- *http://* - which gives the Internet address of a Web page (hypertext URLs);
- *gopher://* - gives the Internet address of a Gopher directory. Gopher is a system used to locate and transfer information that index the filenames in Internet. The syntax of a gopher URL is <gopher://hostname:port/filename>, where *hostname* is the name of the host computer (that usually is a LAN), *port* is the address of his port;
- *telnet://* - allows connect you in real time with another computer in Internet and then to use that computer as you use a local one. For example for networks running under UNIX operating system (and clones), the syntax for telnet:// is telnet:// or [tn3270://](telnet://tn3270://) followed by the name of the computer we want to connect to.
- *ftp://* - which gives the Internet address of a FTP resource. FTP - File Transport Protocol – is the common command set used to upload/download files to/from Web sites.

## URI – Uniform Resource Identifiers

Uniform Resource Identifiers (URI) provide a simple and extensible means for identifying a resource. A URI is a compact string of characters for identifying an abstract or physical resource.

URI's have been known by many names: WWW addresses, Universal Document Identifiers, Universal Resource Identifiers, and finally the combination of Uniform Resource Locators (URL) and Names (URN).

[T. Berners-Lee – RFC 1630] The web is considered to include objects accessed using an extendable number of protocols, existing, invented for the web itself, or to be invented in the future. Access instructions for an individual object under a given protocol are encoded into forms of address string. Other protocols allow the use of object names of various forms. In order to abstract the idea of a generic object, the web needs the concepts of the universal set of objects, and of the universal set of names or addresses of objects.

A Universal Resource Identifier (URI) is a member of this universal set of names in registered name spaces and addresses referring to registered protocols or name spaces. A Uniform Resource Locator (URL), defined elsewhere, is a form of URI which expresses an address which maps onto an access algorithm using network protocols.

URI are characterized by the following definitions for the words that gives his name:

1. *Uniform* - uniformity provides several benefits: it allows different types of resource identifiers to be used in the same context, even when the mechanisms used to access those resources may differ; it allows uniform semantic interpretation of common syntactic conventions across different types of resource identifiers; it allows introduction of new types of resource identifiers without interfering with the way that existing identifiers are used; and, it allows the identifiers to be reused in many different contexts, thus permitting new applications or protocols to leverage a pre-existing, large, and widely-used set of resource identifiers.
2. *Resource* - a resource can be anything that has identity.
 

Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources. Not all resources are network "retrievable"; e.g., human beings, corporations, and bound books in a library can also be considered resources. The resource is the conceptual mapping to an entity or set of entities, not necessarily the entity which corresponds to that mapping at any particular instance in time. Thus, a resource can remain constant even when its content - the entities to which it currently corresponds - changes over time, provided that the conceptual mapping is not changed in the process.
3. *Identifier* - an identifier is an object that can act as a reference to something that has identity.

In the case of URI, the object is a sequence of characters with a restricted syntax. Having identified a resource, a system may perform a variety of operations on the resource, as might be characterized by such words as 'access', 'update', 'replace', or 'find attributes'.

A URI can be classified as a locator, a name, or both:

- the term "Uniform Resource Locator" (URL) refers to the subset of URI that identify resources via a representation of their primary access mechanism (e.g., their network "location"), rather than identifying the resource by name or by some other attribute(s) of that resource.

Although many URL schemes are named after protocols, this does not imply that the only way to access the URL's resource is via the named protocol. Gateways, proxies, caches, and name resolution services might be used to access some resources, independent of the protocol of their origin, and the resolution of some URL may require the use of more than one protocol (e.g., both DNS and HTTP are typically used to access an "http" URL's resource when it can't be found in a local cache).

- the term "Uniform Resource Name" (URN) refers to the subset of URI that are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable.

A URN differs from a URL in that its primary purpose is persistent labeling of a resource with an identifier. That identifier is drawn from one of a set of defined namespaces, each of which has its own set name structure and assignment procedures. The "urn" scheme has been reserved to establish the requirements for a standardized URN namespace.

The following examples illustrate URI that are in common use:

<ftp://ftp.ie.ase.ro/courses/generalinformatics.pdf> - ftp scheme for File Transfer Protocol services  
<gopher://spinaltap.micro.umn.edu/00/Weather/California/Los%20Angeles> - gopher scheme for Gopher and Gopher+ Protocol services

<http://www.math.uio.no/faq/compression-faq/part1.html> - http scheme for Hypertext Transfer Protocol services

<mailto:courseadmin@ie.ase.ro> - mailto scheme for electronic mail addresses

<news:comp.infosystems.www.servers.unix> - news scheme for USENET news groups and articles

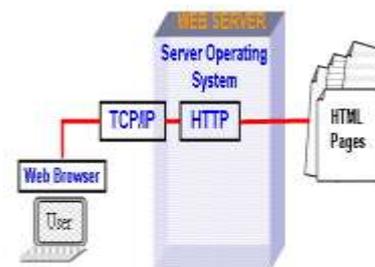
<telnet://melvyl.ucla.edu/> - telnet scheme for interactive services via the TELNET Protocol

## 1.2 Service protocols

The web server, whose base plate described in the §1.1.1, uses at the lower level the HTTP protocol that allows users to request services from his part. A web server may offer a lot of other specialized services, defined as protocols. The HTTP protocol is the core of all that protocols. We introduce in this paragraph these services and the positioning of that ones relatively to HTTP protocol.

### 1.2.1 TCP/IP - HTTP

Figure 1.8 shows the relationship between TCP/IP (see §1.1.1) protocols and HTTP (see §1.1.2) protocol. All user requests addressed to the HTTP server and the responses of this one are send, respectively, received by intermediate of Internet TCP/IP protocols. The common way the user interacts with the web server is the usage of a general web browser. HTTP is the protocol that governs how web browsers (clients) and web servers talk to each other. All messages sent between browsers and servers must be formatted according to the HTTP specification. The HTTP commands allow an application to interpret a page together with his HTML (HyperText Markup Language) links. The HTTP server manages, interprets and acts the HTTP commands.



**Figure 1.8 The positioning of TCP/IP and HTTP protocols**

## 1.2.2 SMTP/POP

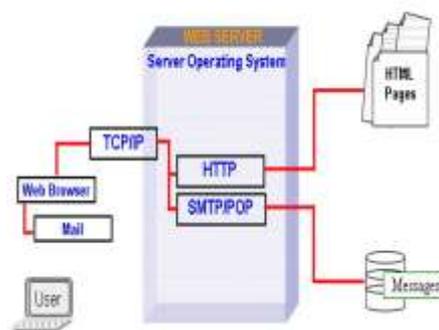
Figure 1.9 shows the position of the mail service and his associated protocols SMTP and POP.

**SMTP** - Simple Mail Transfer Protocol - is an electronic mail protocol that allows mails to travel over the internet.

**POP** – Post Office Protocol – is a utility that allows users to receive their mails.

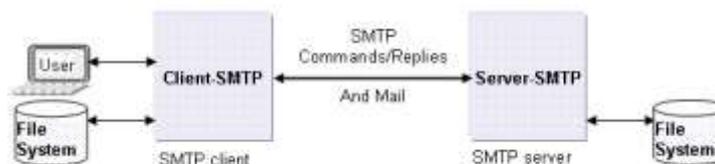
**SMTP**. The objective of the Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently.

SMTP is independent of the particular transmission subsystem and requires only a reliable ordered data stream channel. An important feature of SMTP is its capability to transport mail across networks, usually referred to as "SMTP mail relaying". A network consists of the mutually-TCP-accessible hosts on the public Internet, the mutually-TCP-accessible hosts on a firewall-isolated TCP/IP Intranet, or hosts in some other LAN or WAN environment utilizing a non-TCP transport-level protocol. Using SMTP, a process can transfer mail to another process on the same network or to some other network via a relay or gateway process accessible to both networks. In this way, a mail message may pass through a number of intermediate relay or gateway hosts on its path from sender to ultimate recipient.



**Figure 1.9 The positioning of TCP/IP and HTTP protocols**

The SMTP design can be pictured as in figure 1.10. When an SMTP client has a message to transmit, it establishes a two-way transmission channel to an SMTP server. The responsibility of an SMTP client is to transfer mail messages to one or more SMTP servers, or report its failure to do so. Message transfer can occur in a single connection between the original SMTP-sender and the final SMTP-recipient, or can occur in a series of hops through intermediary systems.



**Figure 1.10 SMTP**

When the user agent on a client host wishes to enter a message into the transport system, it establishes an SMTP connection to its relay host and sends all mail to it. An SMTP client determines the address of an appropriate host running an SMTP server by resolving a destination domain name to either an intermediate Mail eXchanger host or a final target host. An SMTP server may be either the ultimate destination or an intermediate "relay" (that is, it may assume the role of an SMTP client after receiving the message) or "gateway" (that is, it may transport the message further using some protocol other than SMTP). SMTP commands are generated by the SMTP client and sent to the SMTP server. SMTP replies are sent from the SMTP server to the SMTP client in response to the commands. Message transfer can occur in a single connection between the original SMTP-sender and the final SMTP-recipient, or can occur in a series of hops through intermediary systems. In either case, a formal handoff of responsibility for the message occurs: the protocol requires that a server accept responsibility for either delivering a message or properly reporting the failure to do so.

Once the transmission channel is established and initial handshaking completed, the SMTP client normally initiates a mail transaction.

Such a transaction consists of a series of commands to specify the originator and destination of the mail and transmission of the message content (including any headers or other structure) itself. When the same message is sent to multiple recipients, this protocol encourages the

transmission of only one copy of the data for all recipients at the same destination (or intermediate relay) host.

The server responds to each command with a reply; replies may indicate that the command was accepted, that additional commands are expected, or that a temporary or permanent error condition exists.

Once a given mail message has been transmitted, the client may either request that the connection be shut down or may initiate other mail transactions. In addition, an SMTP client may use a connection to an SMTP server for ancillary services such as verification of email addresses or retrieval of mailing list subscriber addresses.

This transmission normally occurs directly from the sending user's host to the receiving user's host when the two hosts are connected to the same transport service. When they are not connected to the same transport service, transmission occurs via one or more relay SMTP servers. An intermediate host that acts as either an SMTP relay or as a gateway into some other transmission environment is usually selected through the use of the domain name service (DNS) Mail eXchanger mechanism.

**POP (Post Office Protocol).** On certain types of smaller nodes in the Internet it is often impractical to maintain a message transport system (MTS). For example, a workstation may not have sufficient resources (cycles, disk space) in order to permit a SMTP server and associated local mail delivery system to be kept resident and continuously running. Similarly, it may be expensive (or impossible) to keep a personal computer interconnected to an IP-style network for long amounts of time (the node is lacking the resource known as "connectivity"). Despite this, it is often very useful to be able to manage mail on these smaller nodes, and they often support a user agent (UA) to aid the tasks of mail handling. To solve this problem, a node which can support an MTS entity offers a maildrop service to these less endowed nodes. The Post Office Protocol - Version 3 (POP3) [RFC1733] is intended to permit a workstation to dynamically access a maildrop on a server host in a useful fashion. Usually, this means that the POP3 protocol is used to allow a workstation to retrieve mail that the server is holding for it. POP3 is not intended to provide extensive manipulation operations of mail on the server; normally, mail is downloaded and then deleted.

Initially, the server host starts the POP3 service by listening on TCP port 110. When a client host wishes to make use of the service, it establishes a TCP connection with the server host. When the connection is established, the POP3 server sends a greeting. The client and POP3 server then exchange commands and responses (respectively) until the connection is closed or aborted.

Commands in the POP3 consist of a case-insensitive keyword, possibly followed by one or more arguments. All commands are terminated by a CRLF pair (Carriage Return and Line Feed). Keywords and arguments consist of printable ASCII characters. Keywords and arguments are each separated by a single SPACE character. Keywords are three or four characters long. Each argument may be up to 40 characters long. Responses in the POP3 consist of a status indicator and a keyword possibly followed by additional information. All responses are terminated by a CRLF pair. Responses may be up to 512 characters long, including the terminating CRLF. There are currently two status indicators: positive ("OK") and negative ("-ERR"). A POP3 session progresses through a number of states during its lifetime. Once the TCP connection has been opened and the POP3 server has sent the greeting, the session enters the AUTHORIZATION state. In this state, the client must identify itself to the POP3 server. Once the client has successfully done this, the server acquires resources associated with the client's maildrop, and the session enters the TRANSACTION state. In this state, the client requests actions on the part of the POP3 server. When the client has issued the QUIT command, the session enters the UPDATE state. In this state, the POP3 server releases any resources acquired during the TRANSACTION state and says goodbye. The TCP connection is then closed.

### 1.2.3 FTP

The FTP (File Transfer Protocol) protocol is used in Internet as a standard for transfer files (for moving files across the Internet). FTP is available as a feature of web browsers for downloading and/or uploading files (figure 1.11).

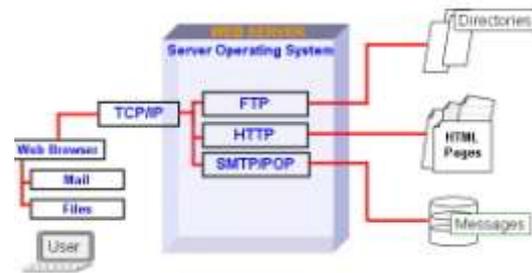


Figure 1.11 The FTP positioning

A FTP site is a server offering libraries of files (images, movies, applications etc). The FTP servers are real mines of freeware (software with no charge for usage) and shareware (applications available at a very lower price) software, images, video, movies, music etc.

The objectives of File Transfer Protocol (FTP), as defined in its specifications, are:

- 1) to promote sharing of files (computer programs and/or data),
- 2) to encourage indirect or implicit (via programs) use of remote computers,
- 3) to shield a user from variations in file storage systems among hosts, and
- 4) to transfer data reliably and efficiently.

FTP, though usable directly by a user at a terminal, is designed mainly for use by programs. Figure 1.12 describes a model for the FTP service, in which:

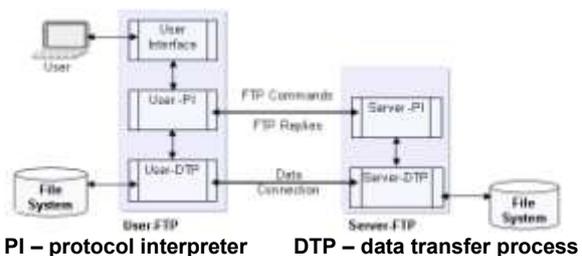


Figure 1.12 The FTP service

- The user and server sides of the protocol have distinct roles implemented in a user protocol interpreter (User-PI) and a server protocol interpreter (Server-PI);

- The user protocol interpreter (User-PI) initiates the control connection from its port U to the server-FTP process, initiates FTP commands, and governs the user data transfer process (User-DTP) if that process is part of the file transfer;

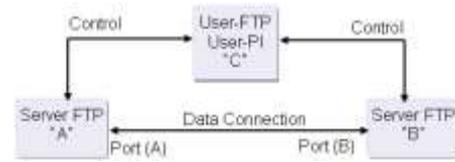
- The user data transfer process (User-DTP) "listens" on the data port for a connection from a server-FTP process. If two servers are transferring data between them, the user-DTP is inactive;

- The server data transfer process (Server-DTP), in its normal "active" state, establishes the data connection with the "listening" data port. It sets up parameters for transfer and storage, and transfers data on command from its protocol interpreter (PI). The DTP can be placed in a "passive" state to listen for, rather than initiate a connection on the data port;

- The FTP commands specify the parameters for the data connection (data port, transfer mode, representation type, and structure) and the nature of file system operation (store, retrieve, append, delete, etc.). The User-DTP or its designate should "listen" on the specified data port, and the server initiate the data connection and data transfer in accordance with the specified parameters. The data port need not be in the same host that initiates the FTP commands via the control connection, but the user or the user-FTP process must ensure a "listen" on the specified data port. The data connection may be used for simultaneous sending and receiving.

In the model described in Figure 1.12, the user-protocol interpreter (User-PI) initiates the control connection. The control connection follows the Telnet protocol. At the initiation of the user, standard FTP commands are generated by the User-PI and transmitted to the server process via the control connection (The user may establish a direct control connection to the server-FTP, from a Telnet terminal for example, and generate standard FTP commands independently, bypassing the user-FTP process). Standard replies are sent from the Server-PI to the User-PI over the control connection in response to the commands.

Telnet allows someone to be on to be on one computer system while doing work on another. Telnet is the protocol that establishes an error-free, rapid link between two computers, allowing you, for example, to log on to your business computer from a remote computer when you are on the road or working from home. Is possible also to log in and use third-party computers that have been made accessible to the public. Telnet uses the computer address you supply to locate the computer you want to reach and connect you to it.

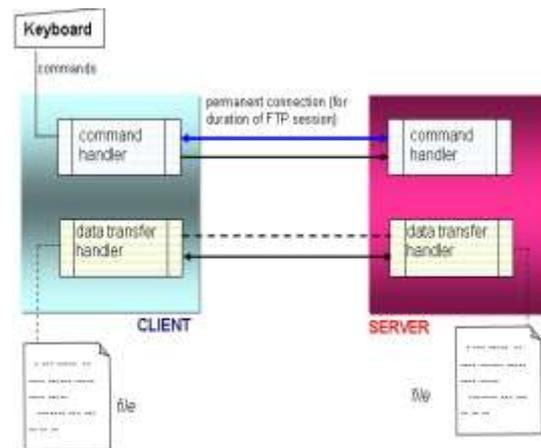


**Figure 1.13 The FTP server-server interaction model**

In the situation a user wish to transfer files between two hosts (neither of which is a local host) the user must sets up control connections to the two servers and then arranges for a data connection between them. In this manner, control information is passed to the User-PI but data is transferred between the server data transfer processes following the model of this server-server interaction as shown in figure 1.13.

### Using FTP line commands

The operating systems offer a tool (named by extension FTP) that allows users to type the FTP commands to the keyboard. By using that tool the user can connect to a FTP server to list the files (dir, mdir and ls commands in figure 1.15), to download files (get for singularly file, mget for many files) or, if it has the necessary rights to upload files (put or mput), to create/delete directories (mkdir, rmdir), change the name (rename) and so on.



**Figure 1.14 The user interaction with the FTP tool**

The way to use the commands and specify their parameters follows the same rules as for the command line commands (MS-Dos prompt option on Start, Programs, Accessories in Windows xx Operating Systems).

Figure 1.14 shows the user model for interaction with the FTP interpreter and figure 1.16 lists the commands available in the Windows Millennium FTP tool.

The virtual file structure that FTP supports is not a general one but cover a wide range of possible files. FTP must consider the following attributes:

- the File Type that can be:
  - character files which contains only characters

```

C:\>ftp 127.1.80.1
ftp> ?
Commands may be abbreviated.  Commands are:
!          delete          literal        prompt        send
?          debug            ls             put           status
append    dir                   mdelete      pwd           trace
ascii     disconnect           mdir         quit          type
bell      get                   mget         quote        user
binary    glob                  mkdir        recv         verbose
bye       hash                  mls          remotehelp
cd        help                  mput        rename
close     lcd                   open         rmdir
ftp> ? mdir
mdir      List contents of multiple remote directories
ftp> quit

```

**Figure 1.15 The FTP tool commands**

(transmissible and printable) such as text files or html documents (ASCII);

- binary files which are considered as stream of bits such as executable files, image files, archive files etc or, in other words, any non text or html file (BINARY);

- the File Structure that can be:
    - unstructured files which are considered as a stream of bytes;
    - record structured files in the case of character files;
  - the Transmission Mode for which a choice can be made for transmitting the file:
    - as a stream of bytes;
    - as a series of blocks of bytes.
- FTP tool requires, when starting, a permanent connection between the command handlers of



**Figure 1.16 The FTP tool commands**

both client and server (figure 1.14). Both client and server process have a separate component that takes care of all component that is responsible for data transfer. When starting FTP session a permanent setup connection is set up the two command handlers. Each time a file is transferred during session a separate connection between data transfer handlers is set and subsequently closed after file transfer has taken place. Figure 1.16 is a snapshot of a ftp session realized in Windows XP environment by using [ftp.exe](#) tool (Start, Run, ftp).

## 1.2.4 NNTP

Network News Transfer Protocol (NNTP) specifies a protocol for the distribution, inquiry, retrieval, and posting of news articles using a reliable stream-based transmission of news among the Internet community. NNTP is designed so that news articles are stored in a central database (News Database - figure 1.17) allowing a subscriber to select only those items he wishes to read. There is a central repository of the news articles in one place (customarily a spool directory of some sort), and a set of programs that allow a subscriber to select those items he wishes to read. The database is provided with indexing, cross-referencing, and expiration of aged messages.

The news server uses a stream connection (such as TCP) and SMTP-like commands and responses. It is designed to accept connections from hosts, and to provide a simple interface to the news database. This server is only an interface between programs and the news databases. It does not perform any user interaction or presentation-level functions. These "user-friendly" functions are better left to the client programs, which have a better understanding of the environment in which they are operating.

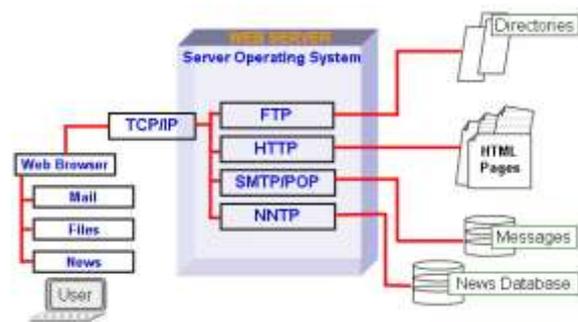


Figure 1.17 NNTP positioning

Using NNTP, hosts exchanging news articles have an interactive mechanism for deciding which articles are to be transmitted. A host desiring new news, or which has new news to send, will typically contact one or more of its neighbors using NNTP. First it will inquire if any new news groups have been created on the serving host by means of the NEWGROUPS command. If so, and those are appropriate or desired (as established by local site-dependent rules), those new newsgroups can be created.

The client host will then inquire as to which new articles have arrived in all or some of the newsgroups that it desires to receive, using the NEWNEWS command. It will receive a list of new articles from the server, and can request transmission of those articles that it desires and does not already have.

Finally, the client can advise the server of those new articles which the client has recently received. The server will indicate those articles that it has already obtained copies of, and which articles should be sent to add to its collection.

In this manner, only those articles which are not duplicates and which are desired are transferred.

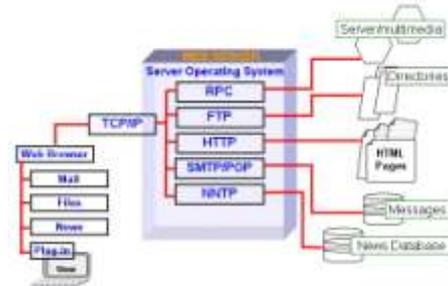
There are popularly two methods of distributing news over Internet: the USENET news system and the Internet method of direct mailing (LISTSERV).

- **USENET newsgroups (Forums)** are worldwide discussion groups in which people share information and ideas on a defined topic. Discussions take place in large electronic bulletin boards where anyone can post messages for others to read.
- **LISTSERV** allows discussions or messaging to be conducted through predefined groups but uses e-mail mailing list servers instead of bulletin boards for communications.

### 1.2.5 RPC and Multimedia

Multimedia is the term used to describe software which (together with appropriate hardware) can interact with user through different techniques such as text, sound, animation or video. The type of hardware required to support multimedia includes sound and video card and capture using microphones, video cameras and scanners. Multimedia software is most common in home computers but also has business applications such as training courses and product promotions. Multimedia functions can be incorporated into both general-purpose software (e.g. word processors and e-mail can include multimedia elements) and application-specific software.

Usually the browser interprets the files with a *html*, *htm*, *gif* or *jpg* extension. If a hypertext link points to a file having another extension such as *pdf*, *mov*, *avi* or *doc* the browser requires an appropriate viewer or reader to read and interpret this file; these viewers are of two categories: plug-ins or add-ons. The **plug-in** is a small program or accessory that can be used to extend a web browser's capability. The difference between them is that the plug-in are integrated into the browser and the user cannot see the difference between this and the browser while the add-on are superposed to the browser for execution. Some of products fits in both categories as for example VDOlive and Real Audio or only in one category, as for example, Flash and Acrobat Reader that are add-on. The multimedia server offers together with the multimedia information the viewers, readers or specific software that can be accessed remotely, via RPC protocol, by the client (figure 1.18).



**Figure 1.18 The RPC protocol**

The remote procedure call (RPC) model is similar to the local procedure call model. In the local case, the caller places arguments to a procedure in some well-specified location (such as a result register) and then transfers control to the procedure, and eventually gains back control. At that point, the results of the procedure are extracted from the well-specified location, and the caller continues execution. The remote procedure call is similar, in that one thread of control logically winds through two processes - one is the caller's process, the other is a server's process. That is, the caller process sends a call message to the server process and waits (blocks) for a reply message. The call message contains, among other things, the procedure's parameters. The reply message contains, among other things, the procedure's results. Once the reply message is received, the results of the procedure are extracted, and caller's execution is resumed. On the server side, a process is dormant awaiting the arrival of a call message. When one arrives, the server process extracts the procedure's parameters, computes the results, sends a reply message, and then awaits the next call message.

The RPC protocol provides the fields necessary for a client to identify itself to a service and vice-versa. Security and access control mechanisms can be built on top of the message authentication.

The RPC call message has three unsigned fields: remote program number, remote program version number, and remote procedure number. The three fields uniquely identify the procedure to be called. Program numbers are administered by some central authority. Once an implementor has a program number, he can implement his remote program; the first implementation would most likely have the version number of 1.

Because most new protocols evolve into better, stable, and mature protocols, a version field of the call message identifies which version of the protocol the caller is using. Version numbers make speaking old and new protocols through the same server process possible.

### 1.2.6 Applications gateways

The Common Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers (figure 1.19). The three words that give the name to the standard describe his functionality:

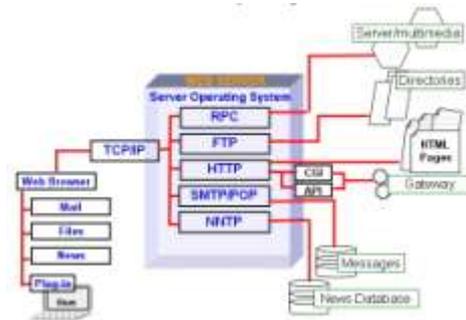
- **Common** specifies a universal method for accessing CGI scripts, that allows to any user, does not mother the used platform, to exchange information with a CGI script;
- **Gateway** defines a bridge between CGI script, Web server and other CGI applications offering the possibility that external programs accept input data and transmit data to other applications;
- **Interface** that reduce the complexity of linking diverse applications to some basic actions describing how external programs can be accessed by clients. For almost Internet users (clients) the process is very simple: the client fill-in the fields of displayed form (by the browser) and press the submit button.

Practically the process follows the steps:

1. The client send data to the Web server;
2. The Web server passes data to a CGI script;
3. The CGI script process data received from server, eventually passes that data to another application and send a response to the Web server;
4. The Web server returns the response to the client (the response can be, for example, the result of querying a database, figure 1.20).

The CGI is a simple interface for running external programs, software or gateways under an information server in a platform-independent manner. The CGI allows an HTTP server and a CGI script to share responsibility for responding to client requests. The client request comprises a Uniform Resource Identifier (URI), a request method and information about the request provided by the transport protocol.

The CGI defines the abstract parameters, known as **meta-variables**, which describe a client's request. Meta-variables contain data about the request passed from the server to the script, and are accessed by the script in a system-defined manner.

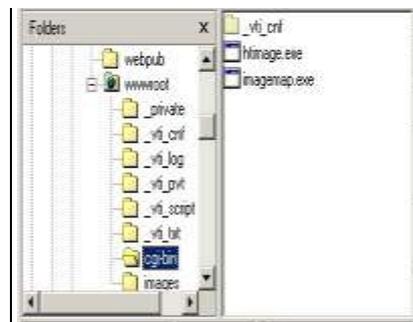


**Figure 1.19 The CGI positioning**

The server is responsible for managing connection, data transfer, transport and network issues related to the client request, whereas the CGI script handles the application issues, such as data access and document processing.

A plain HTML document that the Web server retrieves is static, which means it exists in a constant state: a text file that doesn't change. A CGI program, on the other hand, is executed in real-time, so that it can output dynamic information.

The server acts as an application gateway: it receives the request from the client, selects a CGI script to handle the request, converts the client request to a CGI request, executes the script and converts the CGI response into a response for the client. The script is invoked in a system-defined manner. Unless specified otherwise, the file containing the script will be invoked as an executable program (figure 1.20).



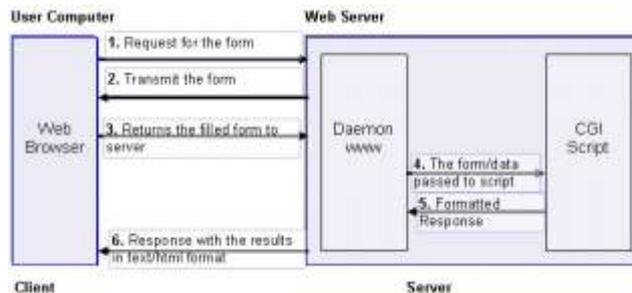
**Figure 1.20 CGI-BIN on a Microsoft Personal Web Server**

A CGI program can be written in any language that allows it to be executed on the system, such as C/C++, Fortran, PERL, TCL, any Unix/Linux shell, Visual Basic, JavaScript etc or, in a scripting language, such as PERL, TCL, or a Unix shell. The scripts must reside in a special directory so that the Web server knows to execute the program (this directory is administered by the webmaster and is called /cgi-bin, figure 1.20).

For example, the Client-Server dialog for processing a form (filled by the user in his browser, for example, like the registration forms displayed by many sites when a user access this for the first time requesting some services requiring authentication) follows the steps (figure 1.21):

1. The Client lunch a URI request that includes a form;

2. The Server receives the request, analyzes that and emits the form to be filled by the Client (user);
3. The Client fill-in the form and send that to Server;
4. The Server runs the CGI script (specified in the message received from Client) that realizes some processing on data contained by the form (e.g. queries a database, computes some values etc) and prepares the obtained results;
5. The CGI script transmits the results to the Server in a Server understandable format;
6. The Server receives the results and sends that to the Client.



**Figure 1.21 The Client-Server dialog when processing a form using CGI scripts**

When a client activates a link to a CGI script the input data are passed to the server. The server associate (and assigns) the transmitted data (stored in meta-variables) with (to) predefined environment variables and verifies if some data are present to the standard input (stdin) device. The environment variables are passed to the CGI script (application). After the CGI script processes data it must return the results (output data) as an answer to client request; as a rule the answer is in a HTML document format that includes a header followed by an empty line and the content (body). The body is formatted accordingly to the information supplied in the header. The statements for server are specified in the CGI header by intermediate of a predefined set of meta-variables whose content defines the request from which:

- 1) *content-type* identifies the MIME data type of the response (e.g.: content-type: text/html);
- 2) *location* for defining the URL of the document returned to the client if the document not directly generated (created) by the script (e.g.: location: http://www.ie.ase.ro);
- 3) *status* providing the status information (error code and explanatory text).

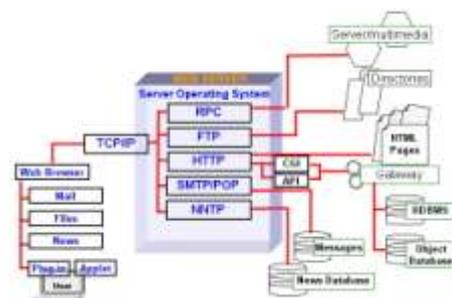
The system software manufacturers offer different alternatives to CGI:

- NSAPI the API alternative to CGI proposed by Netscape for his HTTP servers;
- TSAPI the alternative to CGI proposed for Windows NT servers (Microsoft);
- ODBC (Open DataBase Connectivity) the interface for data access of Windows applications (Microsoft);
- JDBC (Java DataBase Connectivity) the interface for data access of Java applications (Sun);
- SQL (Structured Query Language) the standard language (ISO and ANSI) for data access (initially the SQL used only for relational database model but his extensions allow accessing data managed file managers or database management systems not necessarily for relational model).

### 1.2.7 Applets

An applet is a program written in the Java programming language that can be included in an HTML page (figure 1.22), much in the same way an image is included in a page. When you use a Java technology-enabled browser to view a page that contains an applet, the applet's code is transferred to your system and executed by the browser's Java Virtual Machine (JVM).

Applets have the file extension "class". Some applets consist of more than just one class file, and often other files need to be present for the applet to run (such as JPG or GIF images used by the applet). When you intent to pickup an available applet you must check the documentation for the applet to see if you have all files for it to run. Before embedding an applet on your page you need also to upload the required files to your server.



**Figure 1.22 The applets positioning**

You do not need to know Java to install applets on your pages. There are thousands of free applets available on the internet for almost any purpose. Most of them can be customized without programming. Most of today's browsers can run applets.

An applet can be embedded into a webpage. Usually the applet has several settings that will allow you to personalize it. For instance, if you insert an applet that will work as a menu, you can specify which options should be in the menu, and which pages should be loaded upon click on an option.

Since Java is a real programming language there aren't many limitations to it. Any program running on your computer could possibly have been made as an applet. Spreadsheets, wordprocessors, graphics programs... even entire browsers could be made with Java. However, most applets used on webpages serve much smaller purposes than the ones mentioned. The reason is simple: They need to be transferred through the internet, and therefore can't take up just any amount of space.

When you put an applet on your page you will need to save the applet on your server as well as the HTML page the applet is embedded in. When the page is loaded by a visitor the applet will be loaded and inserted on the page where you embedded it.

### 1.2.8 Wireless Web

**Wireless Web** denote Web based applications enabling users to access digital information from the Internet using wireless mobile communicating devices (such as PDAs, Internet enabled cell phones etc). These devices are characterized by tiny displays screens, low-bandwidth connections and minimal memory. The two main standards governing the Wireless Web for such devices are WAP and I-mode (figure 1.23 a and b).

**WAP** (Wireless Application Protocol) - is a system of protocols and technologies using WML (Wireless Markup Language).

WML is based on XML and optimized for tiny display and was designed only for describing data and not for defining the way data are displayed (as HTML do, for example). To speed the access, each WML file is referred to a 'deck' and consists of several cards that can be displayed sequentially without reconnecting. WAP architecture uses a built-in micro-browser to make a request in WML. The request (figure 1.23 a) is passed to a WAP gateway which retrieves the information from the Internet Server in either HTML or WML format. The Gateway converts HTML to WML so that the client can receive. WAP supports most wireless network standards and operating systems for handheld computing devices such as PalmOS and Windows Mobile 5 for pocket PC, for example.

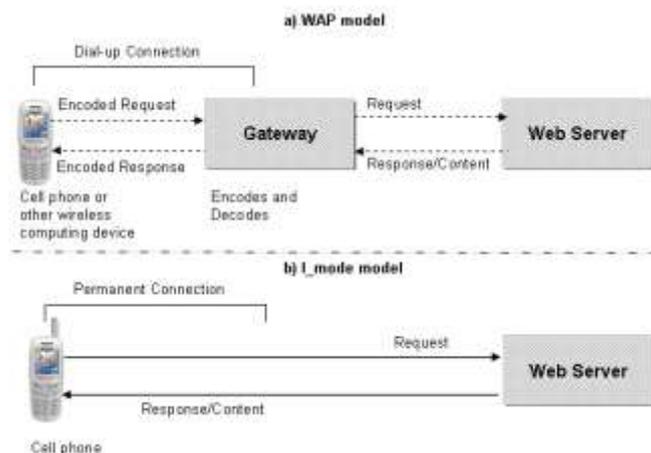


Figure 1.23 WAP and I-mode models

The WAP standard protocols stack looks like:

W	Wireless Application Protocol (WAP)	Internet
	Wireless Application Environment (WAE)	HTML/Java
	Wireless Session Protocol (WSP)	HTTP
	Wireless Transaction Protocol (WML)	
	Wireless Transport Layer (WTLS)	SSL
	Wireless Datagram Protocol (WDP)	

M L	GSM - Global System for Mobile Communication, 9.6-14.4 Kbps HSCSD – High Speed Circuit Switched Data, 38.4-56 Kbps GPRS – General Packet Radio Service, 43-170 Kbps (2.5G) EDGE – Enhanced Data Rates for Global Evolution, 384 Kbps UMTS – Universal Mobile Telecommunication System, 384-2000 Kbps (3G)	TCP/IP
--------	---	--------

(Mobile Access Technologies - the bearers of WML messages)

**I-mode** – is a standard developed by the Japan’s NTT DoCoMo mobile phone network for enabling cell phones to receive Web-based content and services (figure 1.23 b). The I-mode uses a compact HTML to deliver the content that allows easy transformation of traditional HTML pages in the compact HTML. I-mode uses a packet switching technology which allows users to permanently connect to the network and content providers to broadcast relevant information to users.

### 1.3 Web pages, sites and Web browsers – an introduction

#### Web pages and web site - definitions

The documents for World Wide Web (www) are known as Web pages and they are stored on an Internet server and displayed by a Web browser on your computer. Web browsers display Web pages by interpreting the special HyperText Markup Language (HTM or HTML) tags which are used to encode Web pages with display information.

Web pages usually are linked to many different files, such as graphic and multimedia files. You typically keep these files in a folder or set of folders on your disk drive, while you construct your Web site (this folder is known as local web site).

A Web site is defined as a collection of files that are linked to a central Web page, made available via the Web (the pages forms a cohesive collection of information). The Web server is a type of server dedicated to storing, transmitting and receiving the Web pages and Web related files (such GIF and JPEG graphics, AVI sound and images and so on).

The site’s collection of linked files and Web pages are typically tied together into a cohesive collection of information by a home page (generally called default.htm[1], index.htm[1] or simply home.htm[1]). The home page typically contains a topic list which links it to other Web pages in its Web site. All other pages, in a well designed Web site, must offer a button or a link to go back home (or that is provided by the Web browser). When you publish your Web site, you upload the local site folder (and its contents including subfolders) to a Web server, which contains the software that “serves” your Web pages out to Web browsers on computers that are connected to the Internet. Once your local site is published to the Web server it becomes a Web site. The main or home page of the Web site is accessed by using Internet URIs.

#### Web browsers

The www is accessed using a web browser. The interface used by a web browser makes use of hypertext linking techniques. A hypertext is a document that includes highlighted words or phrases. These highlighted sections represent links to other documents or sections of the same document. Clicking the mouse a above one of these links causes it to be activated. A link can be used to move to another document, transfer a file, view a section of video, listen to a sound file or carry out a number of other actions.

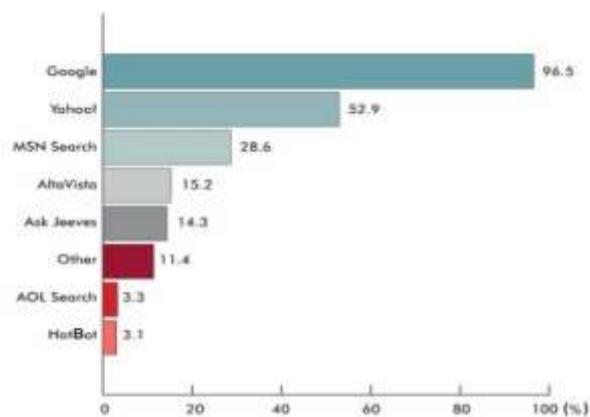
All web browsers provide users with a variety of tools that enable them to navigate through complex collections of WWW pages such as:

- **Navigation buttons** – these allows user to navigate backwards and forwards through the list of pages previously viewed. The browser can also provide additional buttons such as:
  - o *Stop* – for canceling the action currently taken;
  - o *Home* – for returning to the page designated by the user as “main page”;
  - o *Search* – this provides user access to search engines that can be used to locate specific information on the Internet;
  - o *History* – for access to the list of pages previously viewed by the user;

- *Address bar* – for directly entering (typing or choosing from list) the location of a WWW page.
- **Cache** – in order to increase the speed and efficiency the browser can use a temporary storage space to store the copies of any pages the user have viewed (if the user again access later a previously page viewed this one is retriever from that temporary space instead downloading from the original location).
- **Bookmarks** – allows user maintain a directory of web sites (the user can add, edit, delete and organizes addresses);
- **Security** – the modern browsers provide a range of security features, that can be used alone or in combination to obtain varying levels of security, such as:
  - *Digital ID* – provides a means of confirming the identity of a specific user through the use of a small data file called a personal certificate (the file contains encrypted information relating to the user’s identity; that personal certificates are received and send by browser and this one is able to confirm his own identity to a third party or to verify the identity of a third part);
  - *Certificates* – a site certificates contains information regarding the identity of a particular site on the Internet (they encrypted to protect the information they content and used for authenticity check when accessed by browsers);
  - *Ratings* – the ratings used to restrict access to inappropriate contents (such as pornography, for example). The check is based on a defined list of criteria defined by user in the browser to which the site ratings reports. If the site does not meet the criteria the access is denied.
- **Applets** – WWW pages can contain small programs that are activated when a page is accessed. Such programs can take a variety of forms and can include complete, self-contained applications known as applets. These programs are generally considered harmless, they can represent a potential security risk to an organization or individual. For that reason the browser must provide control over the operation of any applets embedded in a www page.
- **Plug-in** – a plug-in is a small program or accessory that can be used to extend the web browser capabilities.
- **Scripts** – all modern web browsers are capable of executing special commands that have been embedded within the body of a www page known as scripts.

In 2005 the classification under the number of peoples using of the first 5 search engines was:

- Google 33%
- Yahoo 31%
- MSN 15%
- AOL Search 10-15%
- Ask Jeevs 5.5%.



**Figure 1.24 Classification of search engines under the preferences when realizing online research (Source: CMO Council, 2005)**

Figure 1.24 shows the classification of search engines under the preferences of users when realizing online research.

### Finding information on the Internet

Information can be found on the World Wide Web in the following main ways:

- By typing in the address bar the URI (URL) of a known web page;
- By using search engines (such as Google, Altavista, AskJeevs etc);
- By using directories / web catalogues / indexes (such as Yahoo);
- By ‘surfing’;
- By intermediate of Web guides (such as [www.about.com](http://www.about.com) and [www.4anything.com](http://www.4anything.com)).

**Web addresses.** The preferred method of reaching a web site is by typing the web address or URL (URI) directly into the web browser (for more information about URL/URI see §1.1.5. For example, by typing the Yahoo URL <http://www.yahoo.com> and pressing Go button the browser will open to you the Yahoo main page.

**Search engines.** The search engines provide an index of all words stored in WWW. Keywords typed by end user are matched against the index and the user is given a list of hyperlinks to pages containing the keywords. By following the hyperlink the user is taken to the relevant web page. One goal of all the search engines is to have the most complete index of files found on the web.

The search engine functionality can be described simply as: the search engine goes out into the Internet, follows the road signs and paths to get where it's going, and collects all of the information in its path. From this point, the information is sent back to a group of servers where algorithms are applied in order to determine the importance of specific documents (to rank the pages and site). Essentially we have an entity that collects data, stores it, and then sorts through it to determine what's important which it's happy to share with others and what's unimportant which it keeps tucked away. Both actions, the search on the web and the discovery of new pages, are realized by automated tools (software packages) called spiders or robots. In all major search engines the spiders crawl from one page to another following the links, as you would look down various paths along your way. An effective crawler needs to be able to index other information, including visible text, alt tags, images and even other non-HTML content such as PDF and word processor documents. Generally, the crawler gets a list of URL's to visit and store that; it does not rank the pages, it only goes out and gets copies which it stores, or forwards to the search engine to later index and rank according to various aspects. Some of the most well known crawlers are Googlebot (Google), MSNBot (MSN), Slurp (Yahoo!) or Teoma (Ask Jeeves). Generally a crawler, when comes to visit a site, checks for a file called "robots.txt" that contains information about which files it can request and which files or directories not allowed to visit.

Most crawling search engines consist of the following main parts:

- *crawler* – a specialized automated program able to follows links found on web pages and to direct the spider by finding new sites for it to visit;
- *spider* – an automatic browser-like program that downloads documents found on the web by the crawler and store them (possibly in a compressed format – Google);
- *indexer* – a program that "reads" the pages that are downloaded by spiders and decides what the page is about and to calculate a quality ranking for each web page (for example, by considering the citations of the page together with the links going out of the page);
- *database* (the "index") – a simply storage of the pages downloaded and processed.
- *results engine* – that generates search results out of the database, accordingly to the user query.

**Google Architecture** (figure 1.25). Most of Google is implemented in C or C++ for efficiency and can run in either Solaris or Linux. In Google, the web crawling (crawler and spider: downloading of web pages) is done by several distributed crawlers (1). There is a URLserver (2) that sends lists of URLs to be fetched to the crawlers. The web pages that are fetched are then sent to the storeserver (3).

The storeserver then compresses and stores the web pages into a repository (4). Every web page has an associated ID number called a docID which is assigned whenever a new URL is parsed out of a web page. The indexing function is performed by the indexer (5) and the sorter (7).

The indexer performs a number of functions: reads the repository, un-compresses the documents, and parses them. Each document is converted into a set of word occurrences called hits. The hits record the word, position in document, an approximation of font size, and capitalization.

The indexer distributes these hits into a set of "barrels" (6), creating a partially sorted forward index (8). The indexer performs another important function: it parses out all the links (9) in every web page and stores important information about them in an anchors file (10). This file contains enough information to determine where each link points from and to, and the text of the link.

The URLresolver (11) reads the anchors file and converts relative URLs into absolute URLs and in turn into docIDs. It puts the anchor text into the forward index, associated with the docID that the anchor points to. It also generates a database of links which are pairs of docIDs. The links database is used to compute PageRanks (12) for all the documents. The URLresolver reads the anchors file and converts relative URLs into absolute URLs and in turn into docIDs. It puts the anchor text into the forward index, associated with the docID that the anchor points to. It also generates a database of links which are pairs of docIDs. The links database is used to compute PageRanks for all the documents.

The sorter takes the barrels, which are sorted by docID (for simplicity purpose but the sort uses many other keys), and resorts them by wordID to generate the inverted index. This is done in place so that little temporary space is needed for this operation. The sorter also produces a list of wordIDs and offsets into the inverted index. A program called **DumpLexicon** (13) takes this list together with the lexicon produced by the indexer and generates a new lexicon to be used by the searcher. The searcher (14) is run by a web server and uses the lexicon built by DumpLexicon together with the inverted index and the PageRanks to answer queries.

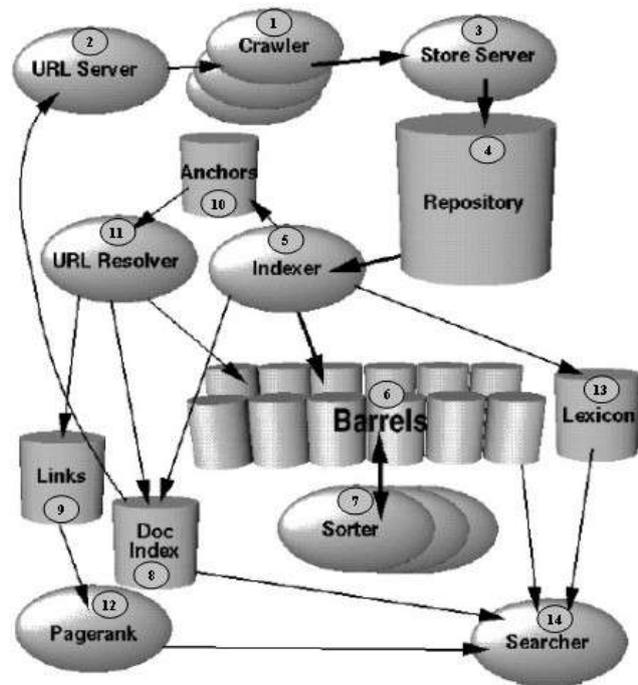


Figure 1.25 High level Google architecture

**Web catalogs or directories.** Web directories provide a structured listing of web sites. They are grouped according to categories such as business, entertainment or sport. In turn each category is subdivided further (for example the sport category subdivided into football, rugby, swimming etc). The web catalogs (such as [www.yahoo.com](http://www.yahoo.com)) work differently from search engines in that they have an hierarchy of information stored under different categories.

A *directory* is used to record information about a particular group of objects. The directory is not intended to be a general-purpose data store. Rather, it is a special type of information repository whose primary purpose is to efficiently store and retrieve information about objects relevant to a particular application or set of applications. A directory service is a physically distributed, logically centralized repository of infrequently changing data that is used to manage the entire environment. Directories are commonly used to store information about users, applications, and network resources such as file servers and printers.

Directories have five important characteristics:

- The storage of information is optimized so that it can be read much more frequently than it is written;
- Information is stored in a hierarchical fashion;
- Information in a directory is attribute-based;
- Directories provide a unified namespace for all resources for which they contain information;
- Directories can efficiently distribute information in a distributed system through replication.

A *directory service* stores and retrieves information from the directory on behalf of one or more authorized users. A traditional directory service provides a means for locating and identifying users and available resources in a distributed system. Directory services also provide the foundation for adding, modifying, removing, renaming, and managing system components without disrupting the services provided by other system components. Today's directory services are used to do the following:

- Store information about system components in a distributed manner. The directory is replicated among several servers so that a user or service needing access to the directory can query a local server for the information;
- Support common searching needs, such as by attribute (for example, "Find the phone number for James Smith") and by classification (for example, "Find all color printers on the third floor");
- Provide important information to enable single-user logon to services, resources, and applications;
- Enable a location-independent point of administration and management. Note that administrative tools do not have to be centrally located and managed;
- Replicate data to provide consistent access. Modifications made to any replica of the directory are propagated around the network so that any application accessing the directory anywhere sees consistent information after the change is propagated.

**Web guides.** The web guides (as for example [www.about.com](http://www.about.com) or [www.4anyting.com](http://www.4anyting.com)) can be considered extensions of web catalogs because they consist of structured information about a particular topic providing articles, definitions, links and news about a particular topic. The web guides are edited by human who will create a structure and rate the information so that only relevant material will be included.

**Other techniques of finding information.** These are represented by the user applications that have an impact on the use of the Internet by organizations such as meta-search tools, offline readers or intelligent agents:

- *meta-search tools* that perform searches across a number of search engines (such as Ask Jeeves ([www.askjeeves.com](http://www.askjeeves.com)) and offers to user more comprehensive and up-to-date lists (the search results lists are collated and processed to remove duplicated items);
- *offline readers* that allows copying individual page, group of pages or entire sites in the local hard drive with preserving the entire functionality (including graphics, animation, scripts and any relevant data) and allowing the user to browse locally the site;
- *intelligent agents* represented by semiautonomous computer programs (as a software 'robot') capable of carrying out one or more tasks specified by user (such as monitoring news and locating the stories of interest to a specific user; searching for a specific product and return details about the manufacturer etc).

#### 1.4 Web services – an introduction

The companies require both hardware and software to realize their processing. Instead of buying and installing software programs they can use Internet or private networks where, by paying a subscription, can rent the same functions from application service providers. An application service

provider (ASP) is a business that delivers and manages applications and computer services from remote centers to multiple users via the Internet or private network. Today's Internet-driven business environment is changing so rapidly that getting a system up and running in three months instead of six could mean the difference between success and failure so that the ASP is a competitive alternative. The ASP enables also small and medium-sized companies to use applications that they otherwise could not afford. The applications offered in that way generally have a proprietary architecture and functionality and the customization do not produce, in all cases, the desired adaptation to the company needs. To avoid all that impediments a new way of deploying and using services emerges, known as Web services.

Web services are software components deliverable over the Internet that enable one application to communicate with another with no translation required. By allowing applications to communicate and share data regardless of operating system, programming language, or client device, Web services can provide significant cost savings, over traditional in-house development.

Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. They perform functions, which can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.

Web Services have emerged as a solution to problems associated with distributed computing. The previous technologies, primarily Common Object Request Broker Architecture (CORBA) and Distributed Component Object Model (DCOM), had some limitations. For example, neither has achieved complete platform independence or easy transport over firewalls. Additionally, DCOM is not vendor independent, being a Microsoft product.

A Web Service forms a distributed environment, in which objects can be accessed remotely via standard interfaces. A Web service performs a specific task or a set of tasks, such as credit card processing, production scheduling, security, third-party billing and payment, for example. Web service uses a three-tiered model, defining three actors: service provider, service consumer, and service broker. This allows the Web Service to be a loose relationship, so that if a service provider goes down, the broker can always direct consumers to another one. Similarly, there are many brokers, so consumers can always find an available one. For communication, Web Services use open Web standards as: HyperText Transfer Protocol (HTTP), Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and the Universal Description, Discovery, and Integration (UDDI) project.

For both service consumer and service broker the Web service is available as an interface that describes a collection of operations that are network-accessible through standardized XML messaging. The Web service is described by using a standard, formal XML notation based on SOAP, called service description. The service description is realized by using WSDL (Web Services Description Language) and provides all of the details necessary to interact with the service, including message formats (that details the operations), transport protocols, and location.

The Web service is created, defined and deployed by **service provider** (figure 1.26). A service provider creates a Web service and its service definition and then **publishes** (1) the service in a service registry (or directory) based on the standard Universal Description, Discovery, and Integration (UDDI) specification.

Once a Web service is published, a **service requester** may **find** (2) the service via the UDDI interface. The UDDI registry provides the service requester with a WSDL **service description** and a URI pointing to the service itself. The service requester may then use this information to directly **bind** (3) to the service and invoke it.

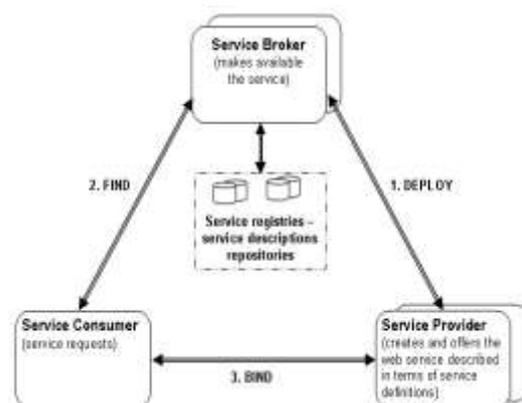


Figure 1.26 Web services model

**SOAP.** Simple Object Access Protocol is a mechanism for sending information in an extensible format. It allows applications to pass data and instructions to one another. SOAP is the envelope

syntax for sending and receiving XML messages with Web services. That is, SOAP is the "envelope" that packages the XML messages that are sent over HTTP between clients and Web services. SOAP can be used to send information or remote procedure calls encoded as XML. A typical SOAP message has the structure:

```
<SOAP:Envelope xmlns:SOAP=http://schemas.xmlsoap.org/soap/envelope/>
  <SOAP:Header>
    <!-- SOAP header go here -->
  </SOAP:Header>
  <SOAP:Body SOAP:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/>
    <!-- SOAP body go here -->
  </SOAP:Body>
</SOAP:Envelope>
```

The SOAP Envelope is used for defining messages. It contains an optional SOAP Header and a SOAP Body. Messages are sent in the SOAP body, and the SOAP head is used for sending other information that wouldn't be expected in the body. For example, if the SOAP:actor attribute is present in the SOAP header, it indicates who the recipient of the message should be. SOAP handles data by encoding it on the sender side and decoding it on the receiver side. The data types handled by SOAP are based on the W3C XML Schema specification. Simple types include strings, integers, floats, and doubles, while compound types are made up of primitive types. Because they are text based, SOAP messages generally have no problem getting through firewalls or other barriers. They are the ideal way to pass information to and from web services.

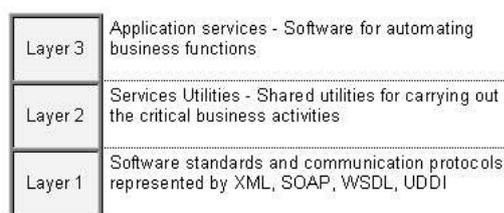
**WSDL.** Web Service Description Language was created to provide information about how to connect to and query a web service. It allows Web service to be described so that it can be used by other applications. The WSDL file defines a service, made up of different endpoints, called ports. The port is made up of a network address and a binding. In turn, the binding identifies the binding style and protocol for each operation.

**UDDI.** Universal Description, Discovery, and Integration standard defines registries in which services can be published and found. It allows Web service to be listed in a directory of Web services so that they can be easily located. The UDDI specification was created by Microsoft, Ariba, and IBM. UDDI defines a data structure and Application Programming Interface (API). The UDDI plays the role of service broker and its function is to enable service consumer.

Web services use a "plug-and-play" like architecture on three layers, as shown in figure 1.27, that differs from the architecture of proprietary applications (such as CORBA or DCOM). These layers are:

- 1<sup>st</sup> layer consists of software standards and communication protocols such as XML, SOAP, WSDL and UDDI allowing information to be easily exchanged between applications;
- 2<sup>nd</sup> layer consists of a service grid to create the environment essential for carrying out critical business activities, utilities for transporting messages, utilities for identifying the available services, shared utilities for security, etc;
- 3<sup>rd</sup> layer consists of application services such as credit card processing, production scheduling or, generally, applications that automates specific business functions.

The companies can use the Web services in conjunction with their existing informatics systems by connecting this to outside services as the needs to do that arises.



**Figure 1.27 Web services architecture**

1 Internet – ARCHITECTURE, OFFERED SERVICES, COMMUNICATION AND NAVIGATION .....	1
1.1 How WANs (and Internet) are organized .....	1
Client/Server Technology .....	6
1.1.1 The Logical Structure of Web Servers.....	10
1.1.2 The “transport” protocols.....	16
1.1.3 The IP addressing.....	16
1.1.4 The DNS .....	19
1.1.5 URL.....	20
URI – Uniform Resource Identifiers.....	21
1.2 Service protocols.....	22
1.2.1 TCP/IP - HTTP .....	22
1.2.2 SMTP/POP.....	23
SMTP .....	23
POP (Post Office Protocol).....	24
1.2.3 FTP.....	25
Using FTP line commands.....	26
1.2.4 NNTP .....	27
1.2.5 RPC and Multimedia.....	28
1.2.6 Applications gateways .....	29
1.2.7 Applets .....	31
1.2.8 Wireless Web.....	32
1.3 Web pages, sites and Web browsers – an introduction.....	33
Web pages and web site - definitions .....	33
Web browsers.....	33
Finding information on the Internet.....	34
1.4 Web services – an introduction.....	37