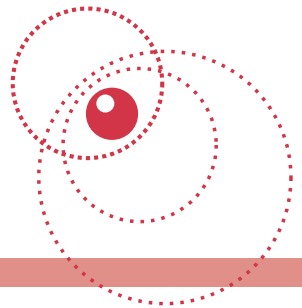


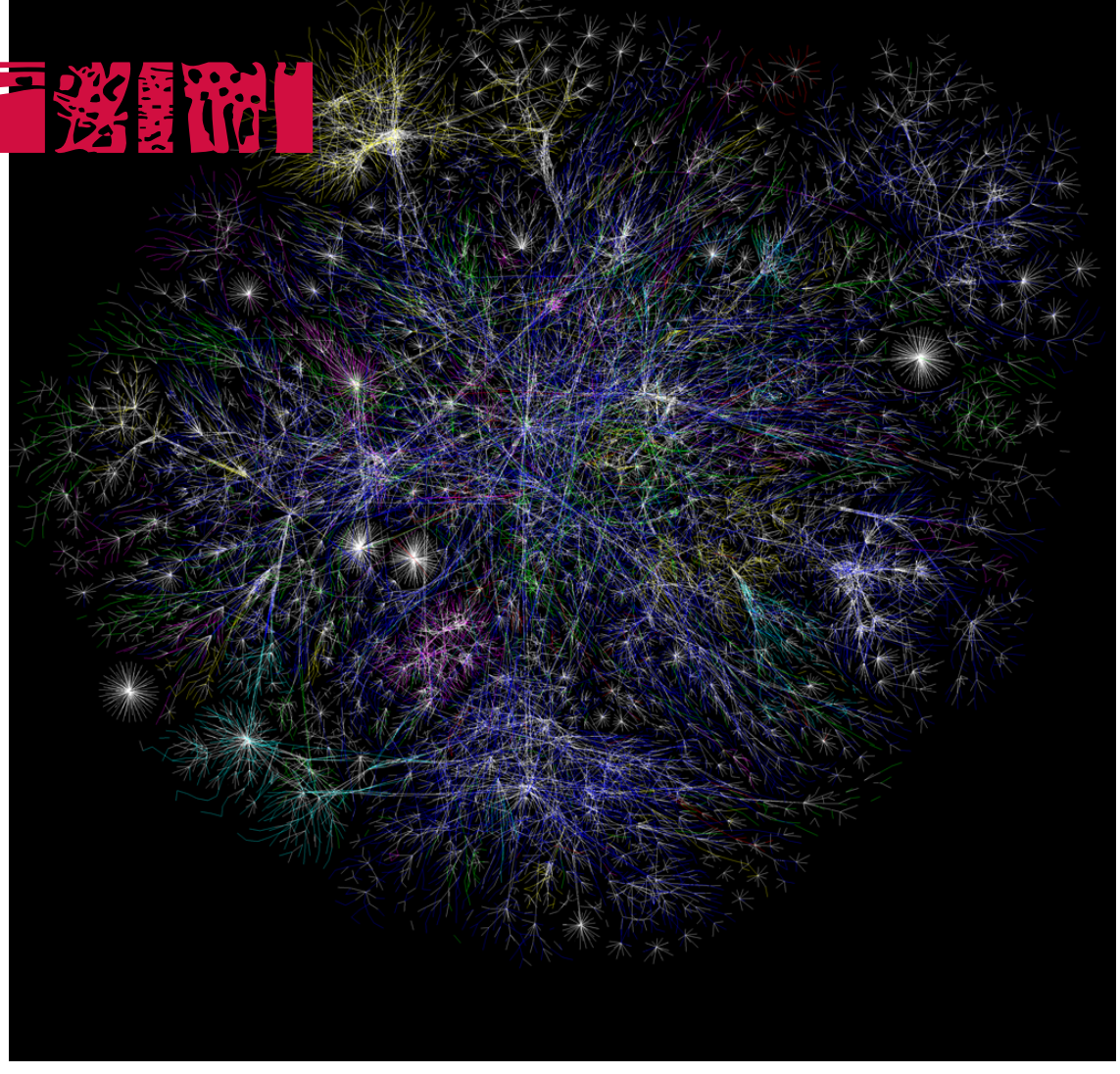
Web Mining

Web technologies: The World Wide Web

27 September 2010

Pierre Senellart

TELECOM ParisTech



http://www.opte.org/

A stack of communication protocols, on top of each other.

Application	HTTP, FTP, SMTP, DNS	
Transport	TCP, UDP, ICMP	(sessions, reliability. . .)
Network	IP (v4, v6)	(routing, addressing)
Link	Ethernet, 802.11 (ARP)	(addressing local machines)
Physical	Ethernet, 802.11 (physical)	

- Addressing machines and routing over the Internet
- Two versions of the IP protocol on the Internet: IPv4 (very widespread) and IPv6 (support still a bit experimental)
- IPv4: 4-byte addresses assigned to each computer, e.g., 137.194.2.24. Institutions are given ranges of such addresses, to assign as they will.
- Problem: only 2^{32} possible addresses (actually, a large number of them cannot be assigned to new hosts, for multiple reasons). This means many hosts connected to the Internet do not have an IPv4 address and some network address translation (NAT) occurs.
- IPv6: 16-byte addresses; much larger address space! Addresses look like 2001:660:330f:2::18 (meaning 2001:0660:0330f:0002:0000:0000:0000:0018). Other nice features (multicast, autoconfiguration, etc.).

- One of the two main transport protocols used on IP, with UDP (User Datagram Protocol)
- Contrarily to UDP, provides reliable transmission of data (acknowledgments)
- Data is divided into small datagrams that are sent over the network, and possibly reordered at the end point
- Like UDP, each TCP transmission indicates a source and a destination port number (between 0 and 65535) to distinguish it from other traffic
- A client usually select a random port number for establishing a connection to a fixed port number on a server
- The port number on a server conventionally identifies an application protocol on top of TCP/IP: 22 for SSH, 25 for SMTP, 110 for POP3...

- IPv4 addresses are hard to memorize, and a given service (e.g., a Web site) may change IP addresses (e.g., new Internet service provider)
- Even more so for IPv6 addresses!
- DNS: a UDP/IP-based protocol for associating human-friendly names (e.g., `www.google.com`, `weather.yahoo.com`) to IP addresses
- Hierarchical domain names: `com` is a top-level domain (TLD), `yahoo.com` is a subdomain thereof, etc.
- Hierarchical domain name resolution: root servers with fixed IPs know who is in charge of TLDs, servers in charge of a domain know who is in charge of a subdomain, etc.
- Nothing magic with `www.google.com`: just a subdomain of `google.com`.

The World Wide Web

- Introduction
- The Web: a market
- HTTP

Internet: physical network of computers (or hosts)

World Wide Web, Web, WWW: logical collection of hyperlinked documents

- static and dynamic
- public Web and private Webs
- each document (or Web page, or resource) identified by a URL

1969 ARPANET (the ancestor of the Internet)

1974 TCP (Vinton G. Cerf & Robert E. Kahn, Turing award winners 2004)

1990 World Wide Web, HTTP, HTML (Tim Berners-Lee, Robert Cailliau)

1993 Mosaic (the first public successful graphical browser, ancestor of Netscape)

1994 Yahoo! (David Filo, Jerry Yang)

1994 Foundation of the W3C

1995 Amazon.com, Ebay

1995 Internet Explorer

1995 AltaVista (Louis Monier, Michael Burrows)

1998 Google (Larry Page, Sergey Brin)

2001 Wikipedia (Jimmy Wales)

2004 Mozilla Firefox

2005 YouTube

Sources: [Electronic Software Publishing Corporation, 2008], [BBC, 2006]

`https://www.example.com:443/path/to/doc?name=foo&town=bar#para`

scheme hostname port path query string fragment

scheme: way the resource can be accessed; generally http or https

hostname: domain name of a host (cf. DNS); hostname of a website may start with `www.`, but not a rule.

port: TCP port; defaults: 80 for `http` and 443 for `https`

path: logical path of the document

query string: optional additional parameters (dynamic documents)

fragment: optional subpart of the document

Relative URLs with respect to a context (e.g., the URL above):

`/titi`	`https://www.example.com/titi`
`tata`	`https://www.example.com/path/to/tata`

- For content: HTML/XHTML, but also PDF, Word documents, text files, XML (RSS, SVG, MathML, etc.)...
- For presenting this content: CSS, XSLT
- For animating this content: JavaScript, AJAX, VBScript...
- For interaction-rich content: Flash, Java, Sliverlight, ActiveX...
- Multimedia content: images, sounds, videos...
- And on the server side: any programming language and database technology to serve this content, e.g., PHP, JSP, Java servlets, ASP, ColdFusion, etc.

Quite complex to manage! Being a Web developer nowadays requires mastering a lot of different technologies; designing a Web client requires being able to handle a lot of different technologies!

- Graphical browsers (cf. next slide)
- Text browsers: w3m, lynx, links (free software, Windows, Mac OS, Linux, Unix); rarely used nowadays
- Other browsers: audio browsers, etc.
- But also: spiders for siphoning a Web site, search engine crawlers (see later on), machine translation software. . .

A very large variety of clients! Web standards (mainly, HTML, CSS, HTTP) are supposed to describe what their interpretation of a Web page should be. In reality, more complex (tag soup).

	Engine	Share	Distribution
Internet Explorer	Trident	50%	with Windows
Firefox	Gecko	30%	Windows, MacOS, Unix FS
Google Chrome	WebKit	10%	Windows, MacOS, Linux FS
Safari	WebKit	5%	MacOS, Windows FC
Opera	Presto	2%	Windows, MacOS, Unix, mobiles FC

FC: free of charge (free as a beer)
FS: free software (free as a man)

Market shares: various sources, precise numbers hard to obtain. IE continually decreasing over the last years.

Trident remains the worst standard-compliant rendering engine.

- Internet Explorer 9 will be released in the coming months, a beta is available.
- Google Chrome has known some success (only two years since its initial release).
- Internet Explorer has three major versions in active use: 6, 7, and 8. For other browsers, the latest major version tend to dominate.
- Most of the current effort of Web browser implementers go these days into improving the speed of JavaScript processors and implementing new standards (CSS 3, HTML 5).

Server	Share	Distribution
Apache	60%	Windows, Mac OS, Linux, Unix FS
Microsoft IIS	30%	with some versions of Windows
lighthttpd	2%	Windows, Mac OS, Linux, Unix FS
Unidentifiable	8%	

- Market share: according to some studies by Opera and Netcraft, precise numbers do not really mean anything.
- Many large software companies have either their own Web server or their own modified version of Apache (notably, GFE/GWS for Google).
- lighthttpd is (as its name suggests!) lighter (i.e., less feature-rich, but faster in some contexts) than Apache.
- The versions of Microsoft IIS released with consumer versions of Windows are very limited.

A large number of different search engines, with market shares varying a lot from country to country.

- At the world level, the big 3:
  - **Google** vastly dominating (more than 90% market share in France!)
  - **Yahoo!** still resists to its main competitor (20% in the US, 50% in Japan)
  - **Bing** (formerly known as MSN, Microsoft Live Search) recent progression (perhaps 10% of the market)
- In some countries, local search engines dominate the market (Baidu in China, Naver in Korea...)

Source (interesting read):
`http://www.search-engine-feng-shui.com/parts-de-marche/`

In July 2009, Microsoft and Yahoo! announced a major agreement:

- Yahoo! stops developing its own search engine (launched in 2003, after the buyouts of Inktomi and Altavista) and will use Bing instead;
- Yahoo! will provide the advertisement services used in Bing.

Transition in progress.

Interestingly, Yahoo! Japan (distinct company from Yahoo! Inc.) is to use Google search engine rather than Bing.

- Application protocol at the basis of the World Wide Web
  - Latest and most widely used version: HTTP/1.1

- Client request:

```
GET /MarkUp/ HTTP/1.1
Host: www.w3.org
```

- Server response:

```
HTTP/1.1 200 OK
...
Content-Type: text/html; charset=utf-8

<!DOCTYPE html ...> ...
```

  - Two main HTTP methods: GET and POST (HEAD is also used in place of GET, to retrieve meta-information only).
  - Additional headers, in the request and the response
  - Possible to send parameters in the request (key/value pairs).

- Simplest type of request.
- Possible parameter are sent at the end of a URL, after a ‘?’
- Not applicable when there are too many parameters, or when their values are too long.
- Method used when a URL is directly accessed in a browser, when a link is followed, and for some forms.

Example (Google query)

URL: `http://www.google.com/search?q=hello`

Corresponding HTTP GET request:

```
GET /search?q=hello HTTP/1.1
Host: www.google.com
```

- Method only used for submitting forms.

Example

```
POST /php/test.php HTTP/1.1
Host: www.w3.org
Content-Type: application/x-www-form-urlencoded
Content-Length: 100

type=search&title=The+Dictator&format=long&country=US
```

- By default, parameters are sent (with GET or POST) in the form: `name1=value1&name2=value2`, and special characters (accented characters, spaces...) are replaced by codes such as +, `%20`. This way of sending parameters is called `application/x-www-form-urlencoded`.
- For the POST method, another heavier encoding can be used (several lines per parameter), similar to the way emails are built: mostly useful for sending large quantity of information. Encoding named `multipart/form-data`.

- The HTTP response always starts with a status code with three digits, followed by a human-readable message (e.g., 200 OK).
- The first digit indicates the class of the response:
  1. Information
  2. Success
  3. Redirection
  4. Client-side error
  5. Server-side error

200	OK
301	Permanent redirection
302	Temporary redirection
304	No modification
400	Invalid request
401	Unauthorized
403	Forbidden
404	Not found
500	Server error

- Different domain names can refer to the same IP address, i.e., the same physical machine (e.g., `www.google.fr` and `www.google.com`)
- When a machine is contacted by TCP/IP, it is through its IP address
- No *a priori* way to know which precise domain name to contact
- In order to serve different content according to the domain name (virtual host): header `Host:` in the request (only header really required)

Example

```
GET /search?hl=fr&q=hello HTTP/1.1
Host: www.google.fr
```

- The browser behaves differently depending on the content type returned: display a Web page with the layout engine, display an image, load an external application, etc.
- MIME classification of content types (e.g., `image/jpeg`, `text/plain`, `text/html`, `application/xhtml+xml`, `application/pdf` etc.)
- For a HTML page, or for text, the browser must also know what character set is used (this has precedence over the information contained in the document itself)
- Also returned: the content length (can be used to display a progress bar)

Example

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 3046
```

- Web clients and servers can identify themselves with a character string
- Useful to serve different content to different browsers, detect robots...
- ... but any client can say it's any other client!
- Historical confusion on naming: all common browsers identify themselves as Mozilla!

Example

```
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; fr;
rv:1.9.0.3) Gecko/2008092510 Ubuntu/8.04 (hardy)
Firefox/3.0.3

Server: Apache/2.0.59 (Unix) mod_ssl/2.0.59 OpenSSL/0.9.8e
PHP/5.2.3
```

- HTTP allows for protecting access to a Web site by an identifier and a password
  - Attention: (most of the time) the password goes through the network uncrypted (but for instance, just encoded in Base64, revertible encoding)
  - HTTPS (variant of HTTP that includes encryption, cryptographic authentication, session tracking, etc.) can be used instead to transmit sensitive data

Example

```
GET ... HTTP/1.1
Authorization: Basic dG90bzp0aXRp
```

- A Web client can specify to the Web server:
  - the content type it can process (text, images, multimedia content), with preferrence indicators
  - the languages preferred by the user
- The Web server can thus propose different file formats, in different languages.
- In practice, content negociation on the language works, and is used, but content negociation on file types does not work because of bad default configuration of some browsers.

Example

```
Accept: text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
```

- Information, as key/value pairs, that a Web server asks a Web client to keep and retransmit with each HTTP request (for a given domain name).
- Can be used to keep information on a user as she is visiting a Web site, between visits, etc.: electronic cart, identifier, and so on.
- Practically speaking, most often only stores a session identifier, connected, on the server side, to all session information (connected or not, user name, data...)
- Simulates the notion of session, absent from HTTP itself

Example

```
Set-Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2;
path=/; domain=.amazon.de;
expires=Fri Oct 17 09:35:04 2008 GMT

Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2
```

- A client can ask for downloading a page only if it has been modified since some given date.
- Most often not applicable, the server giving rarely a reliable last modification date (difficult to obtain for dynamically generated content!).

Example

```
If-Modified-Since: Wed, 15 Oct 2008 19:40:06 GMT

304 Not Modified
Last-Modified: Wed, 15 Oct 2008 19:20:00 GMT
```

- When a Web browser follows a link or submits a form, it transmits the originating URL to the destination Web server.
- Even if it is not on the same server!

Example

```
Referer: http://www.google.fr/
```

The Internet

The World Wide Web

Conclusion

What you should remember

- The Web is not the same thing as Internet!
- Variety of protocols, languages, technologies used on the Web.

- Netcat, a simple command-line utility for establishing TCP/IP connections.
- The Firebug plugin of the Firefox Web browser.

To go further

- Main references:
  - HTML 4.01 recommendation [W3C, 1999]
  - HTTP/1.1 RFC [IETF, 1999]

BBC. Fifteen years of the web. http://news.bbc.co.uk/2/hi/technology/5243862.stm, 2006. Accessed March 2009.

Electronic Software Publishing Corporation. Internet & World Wide Web history. http://www.elsop.com/wrc/h_web.htm, 2008. Accessed March 2009.

IETF. Request For Comments 2616. Hypertext transfer protocol—HTTP/1.1. http://www.ietf.org/rfc/rfc2616.txt, June 1999.

W3C. HTML 4.01 specification, September 1999. http://www.w3.org/TR/REC-html40/.