

# Laboratory exercises for EITF25 Internet - Techniques and Applications

Anders Ardö and Stefan Höst  
Department of Electrical and Information Technology  
Lund University

November 22, 2012

## Contents

<b>1</b>	<b>Protocols</b>	<b>3</b>
<b>2</b>	<b>World Wide Web - WWW</b>	<b>9</b>
<b>3</b>	<b>Infrastructure, applications</b>	<b>17</b>



**LUND INSTITUTE OF TECHNOLOGY**  
Lund University



# 1 Protocols

## 1.1 Objectives

To get a detailed understanding of how an application protocol works. It is exemplified by investigating two common protocols - HTTP and DHCP. Furthermore investigate some security aspects of sending data across wires in a network.

## 1.2 Literature

Khil, Andersson 'Internet': Chapters 3.4, 4.1 - 2, 4.4, 5.1 - 3, 8.1 - 2, 9.2, 9.4

Forouzan 'Data Communications ... 5E': Chapters 1.2, 9.2, 18.4, 24.1-3, 26.1, 26.4

Other places with information:

- HTTP
  - [http://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
  - HTTP Made Really Easy <http://www.jmarshall.com/easy/http/>
- DHCP
  - What is DHCP? <http://kb.iu.edu/data/adov.html>
  - DHCP FAQ: [http://www.dhcp-handbook.com/dhcp\\_faq.html](http://www.dhcp-handbook.com/dhcp_faq.html)
- Wireshark documentation <http://www.wireshark.org/docs/>
  - Internet: [http://www.wireshark.org/docs/wsug\\_html\\_chunked/](http://www.wireshark.org/docs/wsug_html_chunked/)

## 1.3 Home assignments

**Preparation: Read up on all the material in the Literature section and answer all home assignments.**

Answer the following questions:

- What is 'telnet'?
- What is TCP? Describe the frame structure?
- Describe one interaction (one click) between a Web browser and a Web server in terms of messages exchanged.
- What is the relation between TCP and HTTP?
- What is DHCP used for?

## 1.4 HTTP

### 1.4.1 Theory

In order to study your website, you must use a browser (e.g. Firefox, Safari or Internet Explorer). In the browser address bar, type in the address, or as it is called URL (Uniform Resource Locator), to the website you want to visit. URL is a standard that describes how to easily find various resources on a network. A URL starts with the name of the application protocol (or scheme)

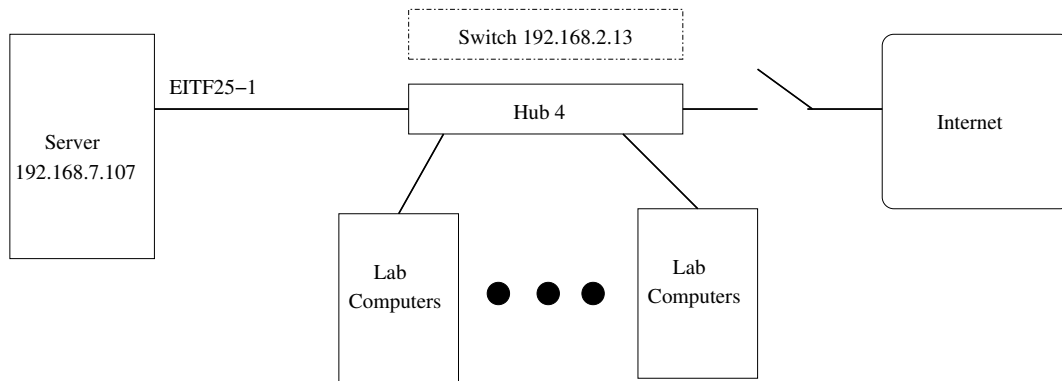


Figure 1: Network configuration during this lab.

followed by a colon (://). Then follow the address to the resource. The address, in turn, consists of two parts; the address (host name or IP number) of the computer and the file name of the resource on that computer.

As an example we consider the url `http://www.libpng.org/pub/png/pngintro.html`. Here the protocol is HTTP, and the host name of the computer is `www.libpng.org` and the path that leads to a file on that computer is `/pub/png/pngintro.html`. Often a default filename is used by the computer, e.g. `index.html` or similar. Then the filename is not needed in the link, as in `http://xkcd.com/773/` where HTTP is used, the computer is `xkcd.com` and the path `/773/`, but here the filename is not revealed to the user.

In full detail every URL consists of some of the following: the scheme name (application protocol), followed by a colon, then, depending on scheme, a host-name (alternatively, IP-no), a port number, the path of the resource to be fetched or the program to be run, then, for programs such as Common Gateway Interface (CGI) scripts, a query string, and with HTML documents, an optional anchor for where the page should start to be displayed. The complete syntax is: `scheme://username:password@hostname:port/path?query_string#anchor`

A computer used to store and deliver web pages is called a web server. A Web server is an application running on your computer, which acts on commands from the browser. To communicate with a web server using web browser by a protocol called HTTP, HyperText Transfer Protocol. This is a set of rules for how communication between a browser and a Web server must be done. When you enter a URL to a web-page and want to see it, the browser is sending a GET command to the web server. For example `GET http://192.168.7.107/whatever.html HTTP/1.0`

#### 1.4.2 Lab assignments

- Login: There is a local account, 'eitf25', with password 'eitf25', on all machines in the lab-room.

Now look at the website at the computer 192.168.7.107 and specifically the file 'demo.html'.

- What is the URL of that website?
- Enter the URL in the browser's address field. The server will now send the HTML document you have written and any images to your browser. The browser then reads the HTML code that you have written and now know exactly how you want your website to look.

Download the page without using a web browser so you can see exactly the information that is sent between the web server and your computer. You will use the `telnet` program. This is a program that you can use to log into another computer from a remote computer. In principle, one can imagine that you are using your keyboard and your screen but the other computer's hard drive.

Open a terminal window (Applications > Accessories > Terminal) and start `telnet` by writing `telnet`. To list available commands write `help`. Quit `telnet` by pressing `Ctrl-C`.

Using `telnet` to test HTTP interaction with a WWW server and retrieve the *entire web page*.

- Which port should you use?
- What does the first line of the returned text mean?
- Why is the connection terminated after one request?

Activate your terminal window and ask 'telnet' to connect to a Web-server: For example:  
`telnet 192.168.7.107 80`

- What does '80' stand for?

You will not receive a response by the web server when you are connected. The screen will be completely black and the web server is waiting for you to send a command. You must type exactly right from the start. If you accidentally make a mistake and then fix it, it will not work. Note that you must only write the file to be retrieved and not the computer name is retrieved from, because you already are connected to the right computer. If the web server disconnects - just connect again.

- Write  
`GET /demo.html HTTP/1.0`  
and press the enter key two times (Why 2?).

If you make a typo, connect again and retype the command. The server will probably time out before you have finished writing, so use copy-past to get it right.

Now you have told the computer that you want to download the file named 'demo.html' in the Web-server root-catalog, just as a Web browser would had said when used that URL. If you have done this correctly now some text should appear.

The text consists of two main parts.

- Identify them and explain the contents. Hopefully you'll recognize the contents of one of the main parts from your browser.

## 1.5 A closer look at the data transfer

### 1.5.1 Theory

In this section we shall study the data transmitted on the link, in our case Ethernet. Images, text and other information that various application programs send between the two computers are transferred using the data link layer. It receives data from the application, plus information, such as what is the type of data and how much data it is. This is encapsulated in a frame. The frame consists partly of data, which is the information received from application software, and

partly by a head containing data link layer information. The head includes information about the sender and receiver.

When frames are sent out the link, it's done in binary form, ie one bit at a time. Between each frame is a space where nothing is sent. In order to study what is sent on the link we need to use a "sniffer program". This is a program that monitors what sent on the link and displays it on screen. The program you will use in this lab is called 'Wireshark'.

### 1.5.2 Lab assignments

We shall take a closer look at what packages are transferred when looking at a website and how these packages look like. Start Wireshark by giving the command `sudo wireshark` in a terminal window. Choose the interface `eth0` from the interface list. Wireshark will then capture all the packets transmitted on the link until you cancel the gathering.

You will see a lot of packets being captured. In order to limit the information on the screen a capture-filter can be used. Press the 'Filter' button (left, near top) and select the display filter 'IP address 192.168.0.1' and modify it to use the IP-no of your computer (can be obtained from the command 'ifconfig' in a terminal-window). Then press OK. Now should only see packets going to and coming from your computer. Start a browser and go to the website When the page comes up, close down your browser and stop the capture. Now all packets that have been transferred appears in the top window. Here you can see the package including transmitter and recipients.

The top window in the column 'protocol', you should look for the first HTTP package that you sent. Highlight it by clicking the line.

- Who is the sender (Source) and recipient (Destination)?
- What does the package contain (seen in the column info)?
- In the bottom panel, you can see exactly the bits transferred in hexadecimal.

In the middle window you can see several different protocols. The most of these we will go through later during the course, but one of the protocols we have reviewed today is HTTP. Click the plus sign in front of the Hypertext Transfer Protocol (HTTP) in the middle window. Now you will see the contents of what has been sent by HTTP, for example which command the browser sent (in this case GET) and what image formats that your browser accepts etc. You can look through the remaining fields in HTTP.

- How many GET did your computer send to the Web server and what is the difference between them?
- Look at the TCP packet surrounding the HTTP packets.
- What function do you think the TCP packets that are sent before and the packets sent after the operation has?

Now click on the packet that contains the web server response. The first packet the Web server response includes information on the version of HTTP you use, what file contains the data (text, image etc.) and its size. Then comes a packet of HTML code. Select this packet in the top pane. Select the field data over HTTP and watch the parts that are marked in the bottom pane. In the left column you can see bits in hexadecimal and in the right column you should reasonably recognize your HTML code.

- Try self to decode of the first hexadecimal characters that have been marked. Are characters encoded using ASCII? ( see `man ASCII`)

## 1.6 telnet vs SSH

In order to this assignment you will have to cooperate with another lab-group.

We will now look at intercepting the communication between two external computers, one lab-computer (not your own) and the server 192.168.7.107.

Start with creating a filter in Wireshark that captures communication from the chosen lab-computer. Start capturing packets, and use telnet on the chosen lab-computer to login to the account 'eitf25', with password 'HemligT' on the server.

### 1.6.1 telnet

Example:

```
$ telnet 192.168.7.107
Trying 192.168.7.107...
Connected to 192.168.7.107.
Escape character is '^]'.
login: eitf25
Password:
```

Stop the capture and log out from the server.

- Analyze the Wireshark capture in order to identify and reassemble the password from the captured network traffic.
- Describe how to extract a password from a captured telnet session.

### 1.6.2 SSH

Do the same but use ssh (Secure SHell) instead of telnet.

Example:

```
$ ssh eitf25@192.168.7.107
Password:
```

- Can you extract a password from a captured ssh session? Why not?

### 1.6.3 DHCP

#### 1.6.4 Theory

DHCP (Dynamic Host Configuration Protocol) is a protocol whereby devices on a network, using DHCP clients, automatically and dynamically can get assigned an IP address and other configuration information. DHCP uses a client-server architecture. The client sends a broadcast request for configuration information. The DHCP server receives the request and responds with configuration information from its configuration database. A device is given a lease on the IP address for a server-specified period of time.

### 1.6.5 Lab assignments

Set up a Wireshark filter to only show bootp/dhcp packets (Filter expression bootp.dhcp - Frame is DHCP).

A procedure to release and reacquire the lease (and thus the connection to network) from the command-line is:

```
sudo dhclient -r eth0
```

```
sudo dhclient eth0
```

Capture all DHCP-packets during releasing and reacquiring your DHCP lease, and answer the following questions:

- Are DHCP messages sent over UDP or TCP?
- Draw a timing diagram illustrating the sequence of packets between your machine (DHCP client) and the DHCP server.
- A host uses DHCP to obtain an IP address, among other things. But a host's IP address is not confirmed until the end of the message exchange. If the IP address is not set until the end of the message exchange, then what values are used in the IP datagrams in the message exchange? For each of the DHCP messages, indicate the source and destination IP addresses that are carried in the encapsulating IP datagram.
- Identify the offered IP address in server packets.

## 1.7 Difference between (Ethernet) hub and switch

### 1.7.1 Theory

You have used Wireshark to sniff packets between two other computers, for example for telnet. This is possible when all 3 computers are connected to a hub. When a hub receives a packet of data at one of its ports from a computer on the network, it transmits (repeats) the packet to all of its ports and, thus, to all of the other computers on the network.

A switch works differently. it automatically divides the network into multiple segments, and supports simultaneous connections of multiple pairs of computers which don't compete with other pairs of computers for network bandwidth. It accomplishes this by maintaining a table of each destination address and its port. When the switch receives a packet, it reads the destination address from the header information in the packet and transmits the packet only on the port where the destination computer is connected. The switch filters and forwards packets between LAN segments. Switches operate at the data link layer (layer 2).

### 1.7.2 Lab assignment

Ask the lab-assistant to move your connection from the hub to the switch.

- Check if you can sniff traffic between two other computers than your lab computer?



## 2 World Wide Web - WWW

### 2.1 Objectives

The purpose of this lab exercise is to improve your understanding of the basic architecture of the World Wide Web (WWW). After the exercise you should be able to create Web pages, apply HTTP to retrieve header lines and entire Web pages. You should also be able to understand how dynamic Web-pages works. In this lab you will learn how to write in HTML language and do a simple website.

### 2.2 Literature

Khil, Andersson 'Internet': Chapter 9.4

Forouzan 'Data Communications ... 5E': Chapter 25.1-2, 26.1

Other places with information:

- A basic introduction into the WWW: [http://en.wikipedia.org/wiki/World\\_Wide\\_Web](http://en.wikipedia.org/wiki/World_Wide_Web)
- HTML, XHTML, CSS
  - HTML:  
<http://www.w3schools.com/html/default.asp>  
<http://www.htmlcodetutorial.com/>
  - CSS:  
<http://www.w3.org/Style/Examples/011/firstcss.en.html>  
<http://www.w3schools.com/css/>
  - XHTML: <http://www.w3schools.com/xhtml/>
- HTTP
  - HTTP made really easy: <http://www.jmarshall.com/easy/http/>
  - Uniform Resource Locator - URL: <http://en.wikipedia.org/wiki/URL>
  - HTTP session management, including cookies:  
[http://www.it.lth.se/courses/will/monash\\_session.html](http://www.it.lth.se/courses/will/monash_session.html)
- PHP
  - PHP tutorial: <http://www.w3schools.com/php/>
  - PHP manual: <http://www.php.net/manual/en/>

### 2.3 Theory

We will in this section write a short HTML document. although several of you probably are familiar with HTML code, here is still a small introduction.

#### 2.3.1 What is HTML?

HTML (HyperText Markup Language) is a language for describing Web pages.

- HTML is not a programming language or protocol, it is a markup language
- A markup language is a set of markup tags

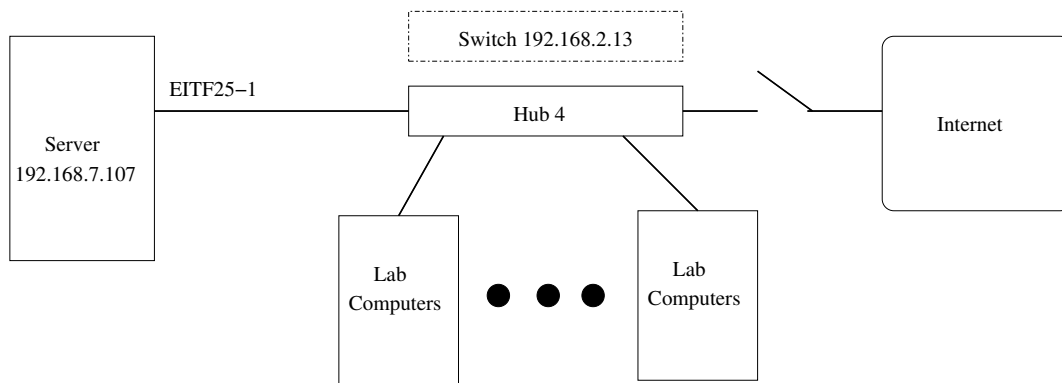


Figure 2: Network configuration during this lab.

- HTML uses markup tags to describe web pages
- A browser, eg Internet Explorer or Firefox, can interpret the HTML and display a Web page in the desired fashion.

HTML markup tags are usually called HTML tags

- HTML tags are keywords surrounded by angle brackets like <html>
- HTML tags normally come in pairs like <b> and </b>
- The first tag in a pair is the start tag, the second tag is the end tag
- Start and end tags are also called opening tags and closing tags

HTML Documents = Web Pages

- HTML documents describe web pages
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages

Example:

```
<h1> This is my homepage. </h1>
```

Here is <h1> start-tag and </h1> end-tag. Tags will in this case mean that the text "This is my homepage." is a heading of type 1. All HTML documents have certain basic standard tags. These are <html>, <head>, <title> and <body> and corresponding end tags. A brief HTML documents with such standard tags may look like below.

Example:

```
<html>
<head>
  <title>First Web-page</title>
</head>
<body>
```

```
<h1> HTML is simple to learn </h1>
All new students are welcome to LTH
<br> Here you will learn a lot!
</body>
</html>
```

Below are the tags used in the document above and a few more tags. It does not matter if you use lowercase or uppercase when writing HTML code. `<h1>` will be interpreted in the same way as `<H1>`.

`<HTML>`

This tag tells your browser that the file contains HTML-coded information. Each HTML document begins and ends with this tag.

`<HEAD>`

HTML documents can be fundamentally divided into two parts, a head and a body. The `<HEAD>` ... `</HEAD>` contains the title, metadata, and information on style sheets and scripts, while the `<BODY>` ... `</BODY>` contains the markup with the visible content.

`<TITLE>`

The title is the name of the page and displayed in the browser's window bar, it does not appear inside the window. The tag `<title>` have to be in the head part of an HTML document.

`<BODY>`

The tag `<BODY>` ... `</BODY>` identifies the body of a HTML document. This part contains the information that the browser will show on the screen.

`<H1>`

HTML has six levels of headings, numbered from 1 to 6, where 1 is the one with the largest font. The text enclosed by `<H1>` ... `</H1>`, is therefore the biggest heading the browser can display. Headings are shown by a browser bold face with a bigger font.

`<BR>`

A browser does not take into account whether you have put carriage returns (CR) in your HTML code. You have to use a tag `<BR>` to break a line and get to the next line. `<BR>` stands for 'BReak'. If we had not used a `<BR>`-tag, the two lines "All new students are welcome to LTH" and "Here you will learn a lot!" would have ended up on the same line, next to each other. You can also mark a paragraph with the tag `<P>` ... `</P>`.

``

With the above command, you can include a picture of your website. Source, 'src', is the path of the source file with the image you want to show. Adding `alt="some text"` in the tag gives the browser a text to display if it can't download your image. Note that the image is not part of the HTML document. Your web browser will download and add the image to the web page built from the code in the HTML document.

Example:

```

```

If the image is in the same directory as the HTML file, you only write the file-name, otherwise you should also specify a path. Since directory structure in the Web server is not the same as on the computer, it's easiest to store the files that make up a web page in the same directory as the HTML document. You can then use so-called relative addressing, and are not dependent on how the other file structure looks like. Relative addressing is when you enter the path to files starting in the directory where the HTML document is.

## 2.4 Home assignments

**Preparation: Read up on all the material in the Literature section and answer all home assignments.**

Answer the following questions:

- What does it mean “stateless”?
- What is a URL and how is it constructed?
- How is an HTML page sectioned?
- What are major elements of the “head” part of a Web page?
- What is metadata?
- What are metatags connected with metadata?
- How can you see metadata in a browser?
- Construct a few HTML Meta-tag lines (assignment 2.5.3) on paper.
- What are the benefits of using XHTML instead of HTML?
- In CSS, what is a 'selector', a 'property', and a 'value'?
- How do you add a special style to an element when you 'mouse' over it?
- Describe the following HTTP commands: GET, HEAD, POST.
- Which are major HTTP return codes and what do they mean?
- What are cookies and what can they be used for?

## 2.5 Lab assignments

### 2.5.1 Login

There is a local account, 'eitf25', with password 'eitf25', on all machines in the lab-room.

## 2.5.2 Web site creation

Start by creating the folder `/var/netwlab/<myWebSite>` (change the folder name '`<myWebSite>`' to something specific for your group) with the command<sup>1</sup>

```
sudo mkdir /var/netwlab/<myWebSite>
```

and make the folder your own with the command

```
sudo chown eitf25 /var/netwlab/<myWebSite>
```

Now you should be able to read and write files in that folder.

Write a simple HTML document and save it as `<me>.html` (change '`<me>`' to for example your first name) in the folder '`<myWebSite>`' in the Web-server root. Make sure that the extension is correct – '`.html`'! Text-editors are available in the menu 'Applications > Accessories' at the top left of the screen.

This HTML document is the foundation of your website. Insert an image in the web-page. Images can be found in the `/var/netwlab/pub/pics` folder. Copy some images to the folder '`<myWebSite>`'.

- Now take a look at how your HTML document looks in a browser. Start Firefox (earth icon in the top bar or via menu 'Applications > Internet') and then open your HTML document entering the URL `http://localhost/myWebSite/<me>.html` in the address bar at the top of the browser.
- Create a small Web site in your folder '`<myWebSite>`' consisting of (at least) two different Web pages (linking to each other), using either XHTML or HTML. The Web site pages should contain at least the following elements: headings, paragraphs, bold text, lists, and both internal (bookmark inside a Web page) and external hyperlinks with anchor-text.
  - What are the URLs of your pages?
  - Check that you can access the pages of another lab group nearby. Lab-machines are numbered 192.168.7.41 – 192.168.7.52.

## 2.5.3 Metadata

- What are metadata used for?
- Add the following metadata elements to a Web page you created: keywords, description, and author.

## 2.5.4 HTML forms

- Add a form to a Web page you created.
- What kinds of control types exist in forms?
- What is 'hidden' used for?

## 2.5.5 Cascading Style Sheets - CSS

Review the CSS tutorial *Starting with HTML + CSS* (from the literature list) at <http://www.w3.org/Style/Examples/011/firstcss.en.html>

---

<sup>1</sup>if the folder already exists you can remove it first with the command '`sudo rm -r /var/netwlab/<myWebSite>`'

- Take one of your pages (with bold-face text) and add an inline CSS style sheet (in the head section) that make bold-face text use a larger font size. The style sheet can look like this

```
<style type="text/css">
b
{
  font-size: 150%;
}
</style>
```

- Change the style sheet to make the bold-face text red on a blue background.
- Add another piece of text in bold-face, but make it belong to a different class than your previous bold text. (Example `<b class="another">more text to view</b>`) Make text like that (but not your other bold-face text) use a smaller font than normal and be blue on a red background.
- Change from an inline style sheet to an external.

### 2.5.6 HTML and CSS validation

- Validate your HTML pages. Use the HTML validator at <http://validator.w3.org/>. Here you can upload your page as a file, and automatically get comments and errors, if any. Make sure there are no errors or comments on your pages.
- You can also validate your CSS code at <http://jigsaw.w3.org/css-validator/>.

## 2.6 Dynamic pages

There are a variety of techniques by which a Web server can generate content dynamically or "on-the-fly" by feeding the output of programs back to the remote browser.

On-the-fly content generation is usually but not always associated with a response to some data previously obtained from the browser via the Common Gateway Interface (CGI) mechanism.

On-the-fly content generation inevitably requires some sort of program to generate HTML. There are numerous possible ways of doing this. In this lab we will briefly look at server pages with PHP.

### 2.6.1 PHP server pages

PHP uses special mark-up within otherwise normal HTML pages. The special mark-up is interpreted as a program by an Apache plug-in and the generated output is inserted into the normal HTML page at that place.

A very simple 'Hi there' page with the wall clock time looks like:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
  <head>
    <title>Hi there</title>
  </head>
  <body bgcolor="#ffffee" text="#000000">
    <h1>Hi there</h1>
```

```

    The time is: <b>
    <?php
        print date('H:i:s');
    ?>
    </b>
</body>
</html>

```

Notice the part in the tags `<?php ... ?>` which is the PHP program generating the dynamic content.

- Copy the example and try it out. Which extension did you use for the filename? Why?
- Write a simple PHP script that add two numbers and displays the result. For your inspiration here is a simple example of a PHP script that uses an HTML form with one parameter.

```

<html>
  <head><title>Simple form</title></head>
  <body>

  <?php
    $yname=$_GET['yname'];
    if ( $yname=='' ) { #empty - no parameters
    ?>

        <form action="simpleForm.phtml" method=get>
        Your name:
        <input type="text" name="yname" size="25" value="">
        <input type="submit" value="Hi">
        </form>

    <?php
      } else { echo "Hi there $yname\n";}
    ?>

    </body>
  </html>

```

If it's called without a the parameter 'yname' it just presents a form that allows you to enter a name and then calls itself but with the parameter set to the name.

If it's called with the parameter 'yname' set it displays a greeting 'Hi there ...'.

- How are parameters transferred (encoded)?

### 2.6.2 Pseudo-state, cookies

Complement your page with a cookie counting the number of times the page is loaded. Cookies can be initialized in PHP before the '`<html>`'-tag as shown below. See also [http://www.w3schools.com/php/php\\_cookies.asp](http://www.w3schools.com/php/php_cookies.asp).

```
<?php
  if (!isset($_COOKIE["kaka"])) {
    setcookie("kaka", 0, time()+3600);
  }
?>
<html>
  <head>
  ...
```

## **2.7 Clean up**

Clean up by removing the folder 'myWebSite' and all it's contents.

## **2.8 Conclusions**

In order to pass the lab, talk to the lab instructor and answer the individual questions included in the lab exercises.



## 3 Infrastructure, applications

### 3.1 Objectives

Study network functions like streaming video using HTTP and service discovery using UPnP. Also have brief look at how to find information on the Internet using two different applications DNS and WWW.

### 3.2 Literature

Khil, Andersson 'Internet': Chapters 4.5, 6.1, 9.7

Forouzan 'Data Communications ... 5E': Chapters 26.6, 28.1-4

Other places with information:

- Digital Living Network Alliance - DLNA <http://www.dlna.org/home>
  - [http://en.wikipedia.org/wiki/Digital\\_Living\\_Network\\_Alliance](http://en.wikipedia.org/wiki/Digital_Living_Network_Alliance)
- Universal Plug and Play - UPnP <http://www.upnp.org/>
  - [http://en.wikipedia.org/wiki/Universal\\_Plug\\_and\\_Play](http://en.wikipedia.org/wiki/Universal_Plug_and_Play)
  - Minidlna - a free UPnP MediaServer <http://minidlna.sourceforge.net/>

### 3.3 Home assignments

**Preparation: Read up on all the material in the Literature section and answer all home assignments.**

Answer the following questions:

- Other carriers  
So far we have been looking at Ethernet based carriers for our network traffic, but there are other possibilities including wireless and other more exotic, see for example the RFC 1149.
- What is a RFC?
- Describe the concept 'routing'.
- How can you find the IP-number of the machine 'www.sunet.se'?
- What is UPnP?
- How does UPnP work?
- What is a media server?

### 3.4 Network load

#### 3.4.1 Theory

Here we will investigate sharing of a network and specifically the capacity. The network for this lab is a 10 Mbps Ethernet (the hub). This means that the maximum transmission speed is 10 Mbps. You are rarely alone on a network which would be required if you want access to the full

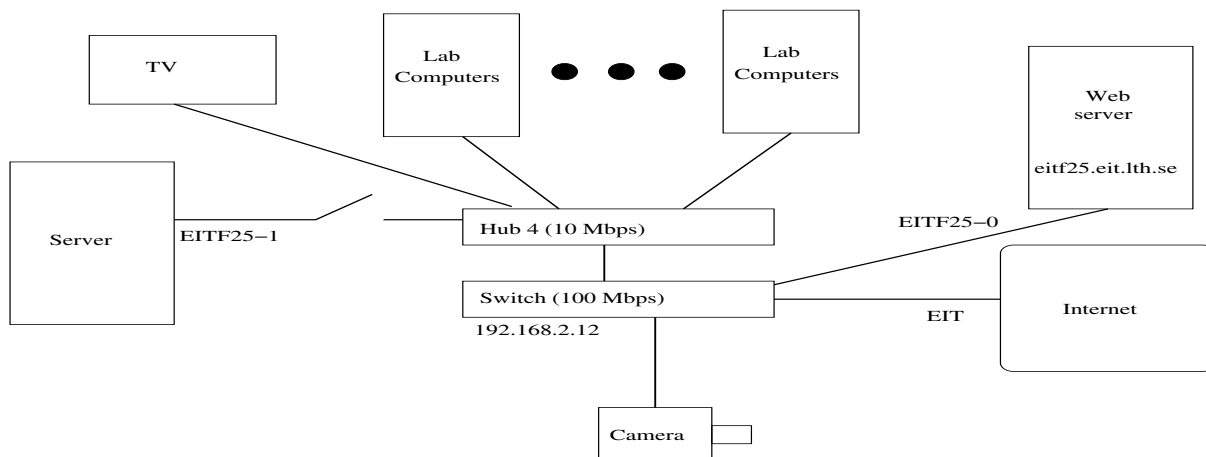


Figure 3: Network configuration during this lab.

capacity yourself. Instead the available bandwidth will be divided between the users who needs it.

To see how this works, you should now try to download a movie from another computer to see how much traffic generated to and from your computer. For this purpose, we have a Web server on a computer which has the name `eitf25.eit.lth.se`.

What should be the theoretical maximum value of download (receiving) bandwidth and under what conditions could we ensure that maximum value?

### 3.4.2 Lab assignments

Go to the server's download page (<http://eitf25server.eit.lth.se/Movies/>) and select the movie, for example `Star.wmv`. Your browser will Open the movie in the "Movie Player". Your computer will now receive a lot of packets containing data that represents the film. You should now look at the traffic to and from your computer. Use a program called System Monitor (menu System > Administration > System Monitor).

Depending on how many people are trying to use the network simultaneously, your receiving speed will vary.

In the Wireshark menu select **Statistics > IO Graphs**. Check that 'Units' is set to 'Bits/Tick' and that 'Tick Interval' is 1 sec. The black line, Graph 1, is default and shows all traffic on the LAN. We will now make Graph 2, the red line, show traffic addressed to and from your computer. Enter the filter `ip.host==<your IP addr>` in the filter box for the red graph. Then click on Graph 2. Now the red graph should show up together with the black. If you are alone on the net the two will overlap.

You can now see your average transfer rate in 'IO Graphs'.

If the movie already is downloaded choose another movie to download.

- What transfer rate to download files (receiving speed) do you have right now?

Now start Wireshark and capture packets for approximately 10 seconds. Has the whole movie already downloaded, choose a different movie from the web server's list. Create again a filter as earlier so that you only see packets to and from your own computer.

You can now see that when your computer has received a certain amount of data from the Web server, your computer sends an ACK packet. ACK is the acknowledgment and used by the receiver to tell the sender that the information is received. Package which fails to arrive, and packages containing any kind of error will be resent until the sender receives an ACK from the receiver. ACK packets is recognized in Wireshark by '[ACK]' in the info field. See for yourself that your computer regularly sends ACK to the computer the movie is downloaded from.

Select one ACK packet in the window at the top. Far to the right it says ACK=number.

- What does the number stand for? Check close by packets to see if your theories are correct.
- If ACKs are not used, what would the effect be?

## 3.5 Routing

### 3.5.1 Theory

For larger networks with several possible paths between computers we need a third kind of device, a router. It chooses the best way to send the packet in to eventually reach its destination.

`traceroute` is a utility that will discover the devices (routers) on a path from one machine to another. It does this through a clever use of the time-to-live (TTL) field in an IP packet's header. The TTL field is used to limit the lifetime of a packet. As a packet passes through a router, the field is decremented. When it reaches zero, a router should discard the packet and send a "time exceeded" error message back to the packet's source. Of course, this error message will have the router's address as its source address. `traceroute` works by sending a series of packets with TTL fields of 1, 2, 3, etc. to the destination. Thus each router along the path will send back an error message saying it discarded a packet. `traceroute` is able to build a list of all the routers on a path to a remote machine by collecting the source addresses from these error messages.

`traceroute` uses an unlikely port number for the destination port so that when the packet finally arrives at its destination, the destination will typically send back an ICMP "port unreachable" error message. Receipt of this message alerts `traceroute` that it has reached the end of the path. Actually, `traceroute` sends packets in sets of threes, each with a different port number, just in case one of the randomly selected ports is actually being used.

### 3.5.2 Lab assignments

Check how `traceroute` is used with the man-command: `man traceroute`

Use `traceroute` to see which routers are between your computer and the following computers:

- `www.eit.lth.se`
- `www.lth.se`
- `www.sunet.se`

## 3.6 Internet@TV

Test out the functionality of browsing the Internet using the TV set (button 'Internet@TV' on the remote control). The TV has the name `tv-2.eit.lth.se` and IP-number `130.235.202.66`.

- What protocols are used? (Hint use Ethernet sniffing with Wireshark)
- Can you reach a random web-server from the user interface on the TV?

## 3.7 DNS

Ask the lab-assistant to move your connection from the hub to the switch.

### 3.7.1 Theory

The Domain Name System (DNS) is a hierarchical naming system for computers connected to the Internet or a private network. It associates various information with domain names assigned to each of the computers. Most importantly, it translates domain names meaningful to humans into the numerical (binary) identifiers associated with networking equipment. An often-used analogy to explain the Domain Name System is that it serves as the "phone book" for the Internet by translating human-friendly computer host-names into IP addresses.

### 3.7.2 Lab assignments

Use tools like ping, host, nslookup, traceroute, whois, GeoIP, and more to find as much information you can about the machines:

- www.lu.se
- www.wikipedia.org

Find latency, geographic location, contact address, how many routers in between, IP-no, etc

## 3.8 DLNA/UPnP

**Find the camera** There is a Web camera connected to the network. Try to find it and have it's images appear in your browser using an URL like `http://XXX/` where XXX should be the IP-no of the camera.

- What strategy can you use to find the IP-no?

Given that you know that the camera uses the protocols SSDP, MDNS and/or IGMP to announce itself, what strategy can be used? (Hint: look up these protocols and see what characteristics they have)

- Show the image from the camera to the lab-assistant!

### 3.8.1 Theory - Service discovery

This type of activity can of course be automated by programs in order to build service discovery tools. It is done within standards/frameworks like DLNA and UPnP.

The Digital Living Network Alliance's (DLNA's) vision/charter is to allow Consumers want to easily and conveniently enjoy digital content using any electronic device, and from any location in their home, and beyond. This will be made possible through the creation of a network of compatible devices that are DLNA certified.

UPnP technology is a key standard required for an interoperable network in the digital home. The term UPnP is derived from plug-and-play, a technology for dynamically attaching devices directly to a computer. UPnP devices are "plug-and-play" in that when connected to a network they automatically announce their network address and supported device and services types, enabling clients that recognize those types to immediately begin using the device.

The UPnP architecture allows peer-to-peer networking of PCs, networked home appliances, and other devices. It is a distributed, open architecture protocol based on established standards such as TCP/IP, UDP, HTTP, XML, and SOAP.

The UPnP architecture supports zero-configuration networking. A UPnP compatible device from any vendor can dynamically join a network, obtain an IP address, announce its name, convey its capabilities upon request, and learn about the presence and capabilities of other devices. DHCP and DNS servers are optional and are only used if they are available on the network. Devices can leave the network automatically without leaving any unwanted state information behind.

### 3.8.2 Lab assignments

Start the 'UPnP Universal Control Point' (menu Applications > Programming > UPnP Universal Control Point). It is a program that discovers and displays which UPnP devices and programs are present on the network.

- Is the Web-camera shown there?
- Are there any other UPnP services/devices?
- Can you see the TV in the list? Why not?

Start the simple UPnP Network Light application and watch it appear in your list. It is a simple UPnP application that can control light intensity of all other instances on the network by UPnP messages.

- Test it by right-clicking on the light bulb and try out the functions in the pop-up menu.

## 3.9 Media-servers/DLNA

Perhaps more interesting is to set up a media-server and test it with a media-player. We will use MiniDLNA as media-server and a standard TV as media-player. MiniDLNA is an open source DLNA/UPnP MediaServer. It allows you to stream your digital media through your home network and listen to/watch it on a variety of DLNA/UPnP compatible devices. MiniDLNA aims to be fully compliant with DLNA/UPnP-AV clients.

MiniDLNA media-server is available on your lab-computer in the directory `minidlna`.

You can start the media-server from the command-line with the command `./minidlna/usr/sbin/minidlna -f ./minidlna/etc/minidlna.conf -R -d` (switches are explained if you try the command `./minidlna/usr/sbin/minidlna -h`).

- Before starting you should edit the configuration file `./minidlna/etc/minidlna.conf` and change the name of your server.
- You might also want to complement the `./minidlna/media/` directory with some media files (videos or photos) of your own.
- Start your media-server with the command `./minidlna/usr/sbin/minidlna -f ./minidlna/etc/minidlna.conf -R -d` and verify that your media-server shows up on the TV (use the button 'Media.P' on the TV remote control).
- Test that you can view the media on your server on the TV (there might be a small latency before your media server appears on the TV).