Florent Mayé (梅羅倫)
A03922203

Electrical Engineering
2014-2015

Technical Essays Assignment 1

Home Network

Introduction

Nowadays, every home has an access to the Internet and more and more connected devices through private networks. But this is in fact only the beginning, many new services are appearing. Some are yet available to people and other are still in development. In this essay, we are going to explore the entire possibilities offer by a home network, explain how they work and exemplify them with my own experience in a DIY private network.

Internet delivery

Often your ISP provides you only one internet connection. But you probably have a lot of devices to connect: computers, smartphones, TV … So you need to create your own personal network in order to give all your devices an Internet access. Then each device will have its personal local IP address in your network, so you will be able to contact specific device. To do that you need to use a router. Indeed from the point of view of an extern device, your network has only one IP, the one given by your internet provider. The solution consists in using a device, commonly called *gateway*, which has interfaces both with your personal network and with the Internet. It will perform *NAT* (*Network Address Translation*): when the gateway receives a query from an intern device to an extern one, then it sends it with its own IP address to the Internet and once the response is back, it transmits the answer of the extern device to the one on the local network.

I made my own Wifi router for my private network using a cheap Linux arm based computer: a *Raspberry Pi* equipped with an antenna. It's really easy, you only have to install/launch few component. First you need to activate *DHCP* service to automatically give an IP to your devices. Then configure your *IP table* to indicate that packets coming from the local interface have to be routed to the Internet and vice versa. Finally activate IP *forwarding* service, so the *Raspberry Pi* changes the IP address of device's packets with its own before sending them to the Internet and retransmit answers to the correspondent device. Now your Gateway with NAT activated is ready. You can choice to activate other services like *local DNS cache* to increase internet performance or firewall for security (I recommend you *shorewall*).

Cloud services

You probably have a lot of devices you use (computers, laptop, tablet …), but nevertheless you always work on the same documents, like for instance your network essay assignment. So you probably would like to access these files without always have to transfer their last version to other devices with a basic USB cable, which is not really convenient. The solution consists in stocking them on a server and synchronizes all your devices with it (which consist of downloading new or modified documents). Recently this trend has increase a lot, and they are lot of companies providing such kind of services on the internet. But often you have to pay, and finally, are you ready to store all your confidential information you don't know were on the Internet, maybe accessible by malicious people who don't care of privacy (advertising companies, NSA ? …) on maybe massively hackable server ? Of course not, and as you have follow networks courses, you are now a resourceful guy, and you are going to set up your own cloud service.

And this is really easily! You only have to install a cloud server on your *gateway* equipped with a hard drives (until now, your raspberry was running on an SD card, which is too small to store data) and the client on all your devices. I recommend you *own* Cloud or *Ajaxplorer* (now called *Pydio*), which are really convenient open source solutions. Configuration is really easy, and you will have access to a graphic administrator web interface. Then all your data are accessible from everywhere(*) with all your devices.

Web server

If you let your raspberry run every time, you can also use it as a personal web server. Installing a LAMP (Linux, Apache, MySQL, PHP) is the most conventional way to do it. Apache is the server process, it will handle all http request. MySQL is a database system, useful if you want to store data, and PHP is a language to program dynamic websites (personally I prefer Python). Then when you put an html or php file in the www folder of your raspberry, it will be accessible with a web browser(*).

Streaming service

You now have a wonderful wireless private network, and an amazing server and cloud service which allow you to access all you document from all around the world(*). But it's not enough for you, you are becoming greedy! You know what to directly access to your home computer from everywhere or see the screen of your smartphone/tablet on your TV! No problem, we are going to directly stream the image of one device screen to another one. Please note that it requires import bandwidth, you won't have any problem on a local network, but if you want to use it outside of your network, it may be lower quality.

This kind of device is also a new trend. Google has already launched its *Chromecast*, which allow to stream devices on your TV. But we don't want to buy it, so we are going to setup this service on our network. If you try to access to an UNIX system (Linux, android, mac …), an easy solution consist in directly use *ssh* sessions to stream back the image of a targeted device. You only need to activate *ssh server* on your targeted device. Then access it using an ssh client with serverX Tunneling activated (*ssh -X*), so the client of the targeted device will send images of applications launched on it. Of course, the end device need to have a serverX running on it (for windows, *Xming* is a good solution). Else you can directly use remote desktop software, which are more convenient. On local network I advise you *VNC* (open source) or *Splashtop* (really efficient). You only need to install the server software on targeted devices and client apps on end devices. But with these solutions, you are not able to access to your devices because they are behind your gateway (or more, else you can do *port forwarding*, see (*)), so client apps cannot contact servers. The solution consists in using a third server which is always connected to the server process. Then when a client tries to contact another device, it will establish a link between them. Two good free software are *Teamviewer* and *chrome remote desktop*. Then your devices are accessible from anywhere *(without *)*. You can now watch a film from your tablet on your TV (if connected to a device with a server process, like a raspberry), or play your favorite PC game on your tablet (of course not while listening a course, particularly a network course!).

(*) Behind the NAT

I said you can access your cloud from everywhere. But actually it is probably false. Why? Because I am a liar. But as I am a liar, the previous sentence was wrong, so in fact I'm not … The fact is that in theory, it should works. But you are probably living in a residence (like a dorm) which has its own private network to provide internet to all residents. So in fact, your gateway is probably behind another gateway. So when a client try to access to the cloud server, the first gateway won't know where to send it.

The only "real" solution is to setup *port forwarding*. With this technic, when a client contact the gateway on a specific port, then the gateway will know that it has to route the query to a specific IP address: yours. To do that, you can try to hack the root account of the gateway to add a port/IP correspondence (really bad solution), or (better solution) to kindly ask your network administrator to do it. If it's not possible, you will need to establish a VPN (*virtual private network)* between your gateway and a third server, then you will be able to access your network through this server. If you don't want to use a third server, there is a last solution. But it's not a really good one because it consists in faking the gateway (she'd better behave herself). The tip is to maintain the gateway in the *"waiting from an answer"* state by sending queries to a nonexistent IP. Then when a client want to connect, it sends fake packet to the gateway using as source IP not its own IP but the nonexistent one the gateway is waiting for. Finally the gateway will route the packet to your gateway and you well be able to establish a connection. Fortunately for you, this method has recently been implemented (called *pwnat*), but it is not widespread because it's not really convenient and not a conventional one.

Conclusion

You now know how to do your own customize private network with a lot of interesting functionalities. Unfortunately, four pages are not enough to talk about all the possibilities of a private network. It also would have been interesting to talk about Bluetooth devices and the new trend of the internet of things.

References

NAT

http://raspberrypihq.com/how-to-turn-a-raspberry-pi-into-a-wifi-router
http://www.revsys.com/writings/quicktips/nat.html
http://www.openbsd.org/faq/pf/nat.html#ipfwd
http://en.wikipedia.org/wiki/Network_address_translation
http://en.wikipedia.org/wiki/IP_forwarding

Cloud

http://owncloud.org ; https://pyd.io

LAMP

https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu

Streaming

https://www.google.fr/chrome/devices/chromecast/#cc-promo
https://www.realvnc.com ; http://www.splashtop.com/personal
http://www.teamviewer.com ; https://support.google.com/chrome/answer/1649523?hl=en

Port forwarding

http://www.fclose.com/816/port-forwarding-using-iptables

VPN

https://openvpn.net/index.php/access-server/docs/admin-guides/182-how-to-connect-to-access-server-with-linux-clients.html

Pwnat

http://samy.pl/pwnat