CSNB113: System Administration
-
12th Topic: Services – Networking Services

Services - Servers

The term server has two meanings: It is the hardware ('machine') that runs as a server. It also is a service that serves the user whenever the user requests a service from the service (server).

A physical machine can easily run a larger number of servcies (servers), and servers can also run on a very normal desktop.

Confused? Naturally!

Again: A server is either a physical or a functional unit.

It depends very much on the context what is meant by the term.

Server and service has the same linguistic root as servant. And in principle there is a reason for this: A server (as a function) "sits there and waits". It waits for a client to connect to it, and demand its services.

A webserver is running, and listening for an incoming request. Then it will reply to that request.

Very practically: A webserver sits and waits for (example) you to connect by typing the address in the address bar of your web-browser. When you press 'Enter', your request is sent to the server, who sits there and waits to return a reply: the webpage that you have requested.

Network Services

In the last lecture, we learned how to configure interfaces, with IP-address and netmask. This is not easy, though system administrators need to know.

Also, when you type http://mail.google.com this is no valid IP-address (compare with the slides of last week ...) .

So we use some servers (services) to help us to configure networking. We use a service to 'know' which IP-address goes with a Domain Name.

Last not least: when you send your request to http://mail.google.com, this request needs to be routed to the webserver of Google, and – of course – the reply (content of the web site) needs to be send (routed) back to you.

All these are done by servers:

configuration: DHCP server

looking up IP-address for a Domain: DNS

directing and sending the packets: routers

DHCP

This is about the most simple of the networking services: It stands for

Dynamic Host Configuration Protocol

When you bring in a new machine into the network, and this machine has no networking information set up (no IP-address), the client (== new machine) will 'ask around' on the local network, if there is a DHCP server, and request an IP-address from it.

If there is a DHCP server, it will offer an IP-address, netmask, and a number of other information to that client. If the client is happy (it usually is), it will use this information, set its interface to the IP-address and netmask as offered, and accept the offer.

Therefore DHCP has 3 basic phases:

request

offer

accept

The IP-addresses that the DHCP offers must be unique (cannot be had by any other machine), and are given out on a temporary basis.

If the client needs / wants the address beyond the expiration time, it will ask for a

renewal

DNS – Type of Service

This is a much more difficult protocol. There are millions of Domain Names in use world-wide, and it is rather complicated on how to retrieve the corresponding IP-address from a domain name.

This task is done by so-called Domain Name Servers.

How would a server in e.g. South America 'know' about the IP-address of http://www.uniten.edu.my? (It actually is 211.24.226.23)

Can all DNS servers everywhere in the world know all domain names? And when you register your own domain, how can all (thousands if not millions) of DNS get to know of this new domain?

This is therefore a hierarchical look-up process:

http://metalab.uniten.edu.my

And this is quite similar to what we saw in the last chapter about a postal address; only it is written on a single line here:

my

edu

uniten

metalab

Therefore, 'my' is considered a top-level domain.

DNS - Recursion

my

edu

uniten

metalab

Sticking to the example: The DNS in South-America doesn't need to 'know' about metalab. It is enough, if it 'knows' a DNS that is able to help it further. → It is sufficient, in South-America, a 'know' the IP-address that knows how to help. (That is, a server responsible for the domains within Malaysia.)

This server in turn, still doesn't know about metalab, but it 'knows' about the lower domains: It might have the IP-addresses of all DNSes in Malaysian educational institutions and organisations ('edu'). And this list then contains the DNS in Uniten, where Uniten staff (ITMS) can add or delete a subdomain ('www', 'metalab', etc.)

Routing

Routing is probably the most difficult task that services on the network need to do. There are jams on networks, sometimes one path is faster than another (depending on daytime or weekday), sometimes political occurrences interest more Internet users than usually, sometimes some routes are out of order.

Therefore, on large networks, routing is done in a dynamical manner, and routes change continuously.

Due to the difficulty of the topic, the details will be explained in future subjects.

What remains to be noted, is that we run a number of servers (services) for all those tasks; we usually run them on servers; not on workstations or desktops.

A DHCP server is usually run on a physical server, a DNS is usually run on a physical server, a machine that implements routing services is usually run on a server (and not on a desktop/workstation).

Services

Of course, all this is done to offer services to the users of the network or the Internet.

The most famous one is the World-Wide-Web (WWW). The client here is known as web browser, while the server is known as web server.

But there are far more:

FTP – File Transfer Protocol

SMTP – Simple Mail Transfer Protocol

SSH – Secure Shell

… and many others

This is a good moment, also under consideration of the lab exercises, to ponder about the differences between server and desktop/workstation/laptop.

While the letter only make sense with a person sitting in front of the keyboard (and mouse), a server can run easily and usefully without any user directly operating it. Rather, a server is usually used (and operated) from afar. Servers also have usually a low-level, low-power graphics card; if at all.

Server Administration Tasks

We discussed these earlier, in the beginning of this course. Now we are much better, have learned a lot more, and can understand much better, that a server is usually run unattended (at least physically unattended: there is often no administrator or operator next to it; nobody directly interfacing the physical machine; often, there is no monitor, at times no keyboard even, and seldom a mouse.)

While a user of a desktop/workstation/tower/laptop can see what the machine actually does, the server has nobody to continuously see its (monitor) output.

Therefore, the server logs are relevant, that note and store and save all relevant occurrences, incidents, possible attacks, usage, even free space on hard disks, logons, reboots, mail being sent and received (not the content, but recipient), success or failure of sending.

It is relevant to have a record (also with respect to quality control) about all happenings on the server, generation and deletion of accounts, amount of data transferred, etc.

Times!

With all of these logs, and services, we understand by now the relevance of the correct times and dates.

While on a workstation this is not so relevant, we need to guarantee this for servers.

And it is not astonishing that there are also time servers available, and we can run time services on our (physical) servers. There are some different time protocols: The early and simple Time Protocol, and the more advanced and more modern Network Time Protocol (NTP).

These services offer a simple client to set the time correctly (Time Protocol); respectively to synchronise continuously with other time servers on the local network and / or the Internet (NTP).

With the more simple Time Protocol, it is still possible to set the time to be correct within one second; which is mostly sufficient.

With NTP it is possible to obtain a precision of the time within less than 50 microseconds off the correct time.
(On Windows this is not guaranteed: http://support.microsoft.com/kb/939322)

Times - Examples

```
$ rdate -p 172.16.0.4
Tue Feb 22 11:56:48 MYT 2011

$ rdate -p 172.16.0.2
rdate: Could not connect socket: Connection refused

$ ntpdate -q 172.16.0.4
server 172.16.0.4, stratum 4, offset -10.981362, delay 0.04201
22 Feb 11:57:35 ntpdate[3041]: step time server 172.16.0.4 offset -10.981362 sec

$ ntpdate 172.16.0.4
22 Feb 11:59:58 ntpdate[3042]: bind() fails: Permission denied
$ sudo ntpdate 172.16.0.4
[sudo] password for user:
22 Feb 11:59:58 ntpdate[3054]: step time server 172.16.0.4 offset -10.985749 sec

$ ntpdate -q 172.16.0.4
server 172.16.0.4, stratum 4, offset -0.000032, delay 0.04205
22 Feb 12:00:08 ntpdate[3056]: adjust time server 172.16.0.4 offset -0.000032 sec
$
```

References

- http://technet.microsoft.com/en-us/library/bb727007.aspx
- http://beginlinux.com/blog/2010/03/making-sense-of-dns/
- http://en.wikipedia.org/wiki/Time_Protocol
- http://en.wikipedia.org/wiki/Network_Time_Protocol
- http://support.microsoft.com/kb/939322