Maxwell

Issue 18.1
November 2014

Stratos II
Reaching beyond

Distributed Hash Tables
An introduction from scratch

Discovery Studytour
Verslag van de studiereis

Ringkerngeheugen als RAM
Het geheugen van vroeger

Electrotechnische Vereeniging

From the Board

President

Dear readers,

My name is Jonas Carpay, and I am the new President of the Electrotechnische Vereeniging. The coming year, I get to use this space to share with you everything that has been going on at the ETV, our board in particular, along with the occasional personal musing. After Bart and I created last years' yearbook, it was clear that we wanted to join the Maxwell. And now we get to do just that, as board members no less!

At the time of writing I am in an aeroplane to Bosnia for this years' EESTEC Chairpersons Meeting, together with Attila, the new commissioner of external affairs. It is our first trip abroad on behalf of the ETV, and we're both very excited to introduce our foreign relations to a healthy dose of Delftsche Gezelligheid. It is one of the many adventures we have had as ETV board members, even though it still feels like we've only just started. For the last few weeks we have spent our days learning the ropes of managing a student society, and our nights visiting other student societies and all manner of parties in our 1997 stretch Cadillac Deville.

One of the most exciting things about a fresh board is wanting to change everything you see, and we are no exception. It is very motivating to look around and think about ways we can improve the ETV and leave our mark on it. That eagerness is a double-edged blade, however. On one hand, change can be welcome when things are the way they are simply because they have always been that way and have now become obsolete or bloated. On the other hand, there are better ways to spend your time than reinventing every wheel you come across, and besides, consistency is always a good thing. There have been quite a few occasions already where we were ready to jump in and start doing things our way, only to find out they had been the way they were for a very good reason.

This Maxwell is an embodiment of that discussion. The new design has cost us a great deal of valuable time, and the old design certainly had its strengths. Was it really worth all that effort? I think it will be, but I will only know whether or not the new design is a success once I get to hold a physical copy. Already it has been a great deal of fun and very educational to be a Maxwell editor.

Ultimately, all these experiences are what being a board member is all about. Meeting new people and flying to countries I've never been to has been a lot of fun, but getting down and dirty, trying to see where we can make a difference and improve the ETV is what I get up for in the morning.

Jonas Carpay
President

Commissaris Onderwijs

Er is dit jaar een nieuw curriculum van start gegaan. De nieuwe eerstejaars merken hier weinig van, maar voor de ouderejaars studenten is er een overgangsregeling opgesteld. Voor de nominaal lopende studenten was dat een kwestie van overstappen op de nieuwe vakken, maar alle studenten met een vertraging moeten een individueel studieprogramma (ISP) opstellen. Dit is het afgelopen kwartaal gebeurd onder begeleiding van een docent.

Wat er ook is veranderd met het nieuwe curriculum, zijn de momenten waarop er getentamineerd wordt. Vanaf dit jaar zijn er van sommige vakken deeltentamens in week 5. Van alle vakken zijn er vervolgens in week 10 ook nog (deel)tentamens. De herkansingen zijn vanaf dit jaar niet meer tijdens de normale tentamens, maar op aparte gelegenheden. De herkansingen van kwartaal 1 zijn in week 2.7 (dat is de week na de kerstvakantie) in de avond. De herkansingen van alle overige kwartalen zijn in de zomervakantie in week 5.3 en 5.4. De reden hierachter is om het doorschuiven van tentamens te voorkomen. Dit is het effect dat studenten hun tentamens niet halen omdat ze in dezelfde week zijn als de herkansingen.

Als er nog vragen, opmerkingen of klachten over het onderwijs zijn, kom alsjeblieft een keertje bij mij langs!

Met resolute groet
Leon Loopik
Commissaris Onderwijs

Maxwell

Colofon
Year 18, edition 1, November 2014
900 copies

Maxwell Committee
Ben Allen
Ludo van den Buijs
Jonas Carpay
Bart Kölling
Dorus Leliveld
Yvo Mulder
Tobias Roest
Marc Zwalua

Contributors
Bart Kölling, Charlotte Treffers, Daniel Kappelle,
Dorus Leliveld, Jan de Jong, Jonas Carpay,
Kees Wissenburgh, Kevin Kouwenhoven,
Leon Loopik, Philip van den Heuvel,
Preethi Pradha Elavarasu,Sjoerd Bosma, Stephen van 't Hof,
Tobias Roest

Advertisements
ASML - inside cover front
OGD - page 16
TenneT- page 29
Pinewood - inside cover back

Printing
Quantes, Rijswijk

Electrotechnische Vereeniging

The 143rd board of the ETV
President: Jonas Carpay
Secretary: Bart Kölling
Treasurer: Sjors Nijhuis
External Affairs: Attila Lengyel
Education: Leon Loopik

Contact
Electrotechnische Vereeniging
Mekelweg 4
2628 CD Delft
The Netherlands

Phone number: +31 (0) 15 278 6189
E-mail: Bestuur-ETV@tudelft.nl
Website: www.etv.tudelft.nl

Subscriptions
Non-members can receive the Maxwell four times a year, against a contribution of €10,- per year.

Change of address
Please send your changes to the address above, or use the website.

Design by:
Jonas Carpay
Tobias Roest

Editorial

Dear readers,

What better way to start a new academic year than with new thoughts? As the freshmen start their new life here in Delft, the Maxwell welcomes a new year full of ideas and reformations. First, I'd like to welcome three new members to the editorial staff. With Bart, Dorus and Jonas as brand new editors, the Maxwell is sure to keep getting better and better, and this first edition is proof of that statement. Being the first edition of the new year, we really wanted to make some big improvements. One that immediately catches the eye is the new design. Over the last couple of weeks we have been very busy redesigning the Maxwell from scratch. Being the first edition with this new look, the design is still a work in progress. Over the next couple of issues we will keep updating the design to make it as good-looking as possible. Furthermore, we decided to no longer give each issue a central theme. Instead, we switched to having a cover story for every issue. The cover story for this issue is about the DARE rocket launch was on everyones mind last october. Furthermore, we have a new column. Every issue, we will highlight one piece of the studieverzameling in a small article. Another new column is the interview with a startup company in Delft. The old ones, such as (among others) From the Board and the career column will still be here as well, although they may have changed a bit. These are the biggest changes in this edition, but be sure to keep an eye out for more new stuff!

Tobias Roest

Distributed Hash Tables

An introduction to DHT's

Freedomlayer.org

I want to begin with an example from life. You might want to read it even if you have some general knowledge about DHTs, because it might give you some new ideas about where DHTs come from.On your cellphone, most likely you have a list of contacts. Could you maintain contact with all your friends without having this list? More specifically - What if every person in the world could remember only about 40 phone numbers. Given that structure, could we make sure that every person in the world will be able to call any other person in the world? In the spirit of no hierarchical related solutions, we will also want to have a solution where all the participants have more or less symmetric roles.dolore inis aboratur?

First solution - Phone ring

General structure

A simple solution would be as follows: We sort the names of all the people in the world into a very big list. (Assume that people have unique names, just for this article). Next, every person will have the responsibility of remembering one phone number: The phone number of the next person on the list.

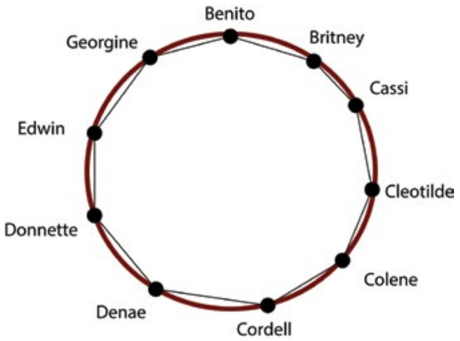


Figure 1: The phone list drawn as a ring, with lines representing the connection between people on the list.

As an example, if the list is as follows:

1. Benito Kellner
2. Britney Antonio
3. Cassi Dewolfe
4. Cleotilde Vandyne
5. Colene Kaufmann
6. Cordell Varley
7. Denae Fernandez
8. Donnette Thornberry
9. Edwin Peters
10. Georgine Reneau

Then Britney will keep the phone number of Cassi. Cassi, in turn, keeps the phone number of Cleotilde. Cleotilde keeps the phone number of Colene, and so on.

The list is cyclic. You can think of it as a ring, more than as a list. The last person on the list will remember the phone number of the first person on the list. (In our list, it means that Georgine keeps the phone number of Benito).

Now assume that Benito wants to call Edwin. How can he do that? He will first call Britney, because he knows her phone number. He will ask Britney for the name and phone number of the next person on the list. That would be Cassi.

Next Benito will call Cassi, and ask her for the name and phone number of the next person on the list. That would be Cleotilde. At this point Benito can forget the name and phone number of Cassi, and move on to calling Cleotilde. Benito will keep advancing in the list until he finally finds Edwin.

We call this operation of finding someone on the list a query, or a search.

Joining the ring

Assume that some person **X** wants to join the phone list. How can we add **X** so that the structure is preserved?

X will first contact some person **Y** that is already on the list. Let us assume that **X** contacts Denae for example. Denae will then search for a suitable place for **X** on the cyclic list, so that the list will stay sorted. If in our example **X** is Gary Jablonski, Then Denae search will yield that Gary should be put between Edwin and Georgine.

After **Y** Finds a place for **X** on the list, **Y** will tell **X** about his designated location in the list. Then **X** will join the list at this place. (We assume that **X** is a good per-

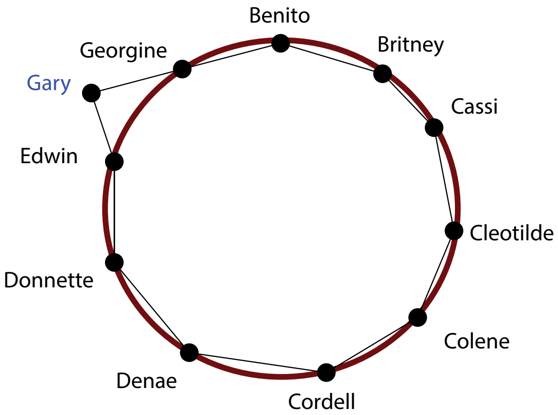


Figure 2: The new state of the list, after Gary has joined.

son, and he will just go to his designated place without giving us any trouble.)

Following our example of Gary Jablonski joining the list, the new list will look somehow like this:

1. Benito Kellner
2. Britney Antonio
3. Cassi Dewolfe
4. Cleotilde Vandyne W
5. Colene Kaufmann
6. Cordell Varley
7. Denae Fernandez
8. Donnette Thornberry
9. Edwin Peters
10. Gary Jablonski
11. Georgine Reneau

Of course that in the new setting, Edwin for example now has to remember only Gary's phone. He shouldn't keep remembering Georgine's phone number, because it is not needed anymore.

(The new state of the list, after Gary has joined.)

Analysis

Whenever person **A** wants to find person **B** on the list, he will have to traverse the list of people one by one until he finds **B**. It could take a very short time if **A** and **B** are close on this list, however it could also take a very long time if **A** and B are very far (In the cyclic sense. In the worst case, **B** is right before **A** on the list).

However we could find the average time it takes for **A** to contact **B.** It would be about n/2, where n is the amount of people on the list.

In addition, we can also measure the amount of memory used for each of the people on the list. Every person is responsible for remembering exactly one people's name and phone number. (The next one on the list).

Whenever a person wants to call someone, he will have to remember an additional phone number, which is the next person he is going to call. This is not much to remember though.

In more mathematical terms, we say that a search (or a query) costs **O(n)** operations, and every person on the list has to maintain memory of size **O(1)**.

Joining the network also costs **O(n)** operations. (That is because joining the network requires a search).

Improving search speed

So far we managed to prove that we could live in a world without contact lists. We just have to remember a few names and phone numbers (In the simple solution above: only one name and one phone number) to be able to call anyone eventually. Though "eventually" is usually not enough. We don't want to call half of the world to be able to contact one person. It is not practical.

Just imagine this: Every time that someone in the world wants to call someone else, there is a probability of 1/2 that he will call you on the way! Your phone will never stop ringing.

What if we could somehow arrange the phone list so that we will need to call only a few people for every search? Maybe if we remember a bit more than one people's phone number, we could get a major improvement in search performance.

Adding more immediate links

A first idea for improving the phone list would be that each person will remember more of his list neighbours phone numbers. Instead of remembering just the next on the list, why not remember the two next people on the list?

In this structure, every person has to remember 2 names and phone numbers, which is not so much more than the 1 that we previously had. However, the improvement in the search operation is major: A search operation will now cost an average of n/4 operations, instead of n/2 that we had previously. (Implicitly, it also improves the cost of joining the network).

We can add more and more records to remember for each of the people on the phone list, to get further improvement in the speed of one search operation. If each person on the list remembers **k**neighbors forward on the list, then the search operation will be k times faster. As **k** can't be so big (Generally we will assume that people on the list can not remember more than **O(log(n))** stuff), we can only get so far with this method.

Maybe if we choose to remember only specific people on the list in some special way, we could get better results.

Chord

So far we have discussed a very nice phone list game, and you might not understand why care about it at all. Let me formulate the question differently.

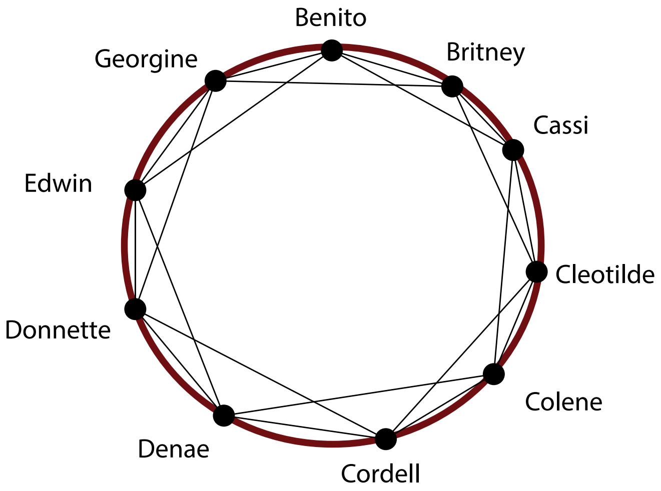


Figure 3: The list with **k = 2**. Search operation is twice as fast.

Assume that we have a set of n computers, or nodes, connected to the Internet (The good old internet that you know and use). Each computer has some kind of unique name. (The unique name is not his Internet Address.)

We want to create a communication structure (Or an overlay network) that satisfies the following requirements:

- Each computer will able to "contact" each of the other computers.
- Every computer can remember the addresses of only about **O(log(n))** other computers' addresses.
- Computers might join or leave the network from time to time. We would like to be able to allow that while preserving the general structure of the network.

Before dealing with solving this problem, I want to discuss some of the requirements. Lets begin with the first requirement. What does it mean to be able to "contact" other computers? Let me give you a simple use case. Lets assume that every computer holds some chunk of information, some kind of a very big table. Maybe this table is a distributed database. Maybe part of a file sharing protocol. Maybe something else. We want to make sure that every computer can reach any other computer, to obtain data for example.

Regarding the second requirement - Every computer can remember only a few addresses. Why can't every computer keep the addresses of all the other computers? Well, there are a few practical reasons for that. First - There might be a lot of computers. n might be very large, and it might be heavy for some computers to remember a list of n addresses. In fact, it might be more than remembering n addresses. A TCP connection between two computers, for example, has to be maintained somehow. It takes effort to maintain it.

But there is another reason. Probably a more major one. We want that this set of computers will be able to change with time. Some computers might join, and others might leave from time to time. If every computer is to remember all the addresses of all the other computers, then every time a computer joins this set, n computers will have to be informed about it. That means joining the network costs at least **O(n)**, which is unacceptable.

If we want computers in this set to be able to bear the churn of computers joining and leaving, we will have to build a structure where every computer maintains links with only a small number of other computers.

Adapting the phone ring solution

As you have probably noticed, this problem is not very different from the phone list problem. Just replace Computers with People, Computers' unique identities with the people's unique names, and Computer's Internet Addresses (IPs) with People's phone numbers. (Go ahead and do it, I'm waiting)

To the solution for the Computer's case is as follows: First we sort the node's names somehow. (If the nodes' unique names are numbers, we just use the order of the natural numbers). Then we build a ring that contains all the nodes, ordered by their name. (We just think about it as ring, we don't really order the nodes physically in a ring, just like we didn't order the people in a circle when we dealt with the phone list problem)

Every node will be linked to the next node on the ring. Searching a node (By his unique name) will be done by iteratively asking the next node for the name and address of the next next node, until the wanted node is found.

Joining the network is as described in the phone list case. (Leaving the network is a subject we will discuss in a later time.)

Here, just like in our description of the previous problem (The phone list), we could also improve the speed of search if every node will keep more links to direct neighbours. However, as we have seen before, we can only get so much improvement in this method, and we would like to find a better idea for link structures between the nodes.

Improving the Search

The following leap of thought could be achieved in more than one way. One way to begin with it to think ituitively about how we manage to find things in the real world.

Intuition from real world searching

Lets assume that you want to get to some place, and you are not sure where it is. A good idea would be to ask someone how to get there. If you are very far from your destination, most likely the person you asked will give you a very vague description of how to get there. But it will get you starting in the correct direction.

After you advance a while, you can ask somebody else. You will get another description, this time more a detailed one. You will then follow this description, until you get closer.

Finally when you are really close, you will find someone that knows exactly where is that place you are looking for. Then your search will end.

This might lead us to think that maybe the network of links between nodes should be arranged as follows:

- Every node **X** is "linked" to nodes with names closest to his name. (His two immediate neighbors on the ring, for example).
- Every node **X** is connected to other nodes from the ring: As the distance **X** becomes greater, **X** is connected to less and less nodes.

Generally: X knows a lot about his close neighbourhood, however he knows little about the parts of the rings that are far.

Binary Search

A different way to look at the search problem is from the angle of a more common method: Binary search. Given a sorted array, we could find an element inside the array in **O(log(n))** operations, instead of the naive **O(n)**.

How could we apply Binary Search to our case? In the binary search algorithm in every iteration we cut the array to two halves, and then continue searching in the relevant half. We can do that because we have random access to the elements of the array. That means - We could access any element that we want immediately. We could access the middle element immediately.

In the simple ring setting (Every node is connected to the next and previous nodes) we don't have random access. However we could obtain something similar to random access if we added the right links from every node. Take some time to think about it. How would you wire the nodes to obtain the "random access ability"?

Binary search Wiring

To explain the next structure of links I want to discuss some notation stuff first.

We assume that the names of all the nodes are numbers that could be represented using s bits. In other words, the names of nodes are from the set: $\mathbf{B_s:=\{0,1,2,\dots,2s-1\}}$. The details here don't really matter. All that matters is that $2^s \geq n$, so that there are enough possible unique names for all the nodes in the network.

We also want to treat the set $\mathbf{B_s}$ as cyclic modulo 2^s.

Let x be some node on the ring. (x is the name of this node. $x \in \mathbf{B_s}$). We will connect x to the following nodes on the ring:

- [x+1]
- [x+2]
- [x+4]
- ...
- [x+2^{s-1}]

The notation [y] means the first node that his name is bigger than y.

In the picture: The ring represents the set Bs of possible names for nodes. (With s=6). Blue points are existing nodes. Their location on the ring represents their name. Cuts on the ring represent the exact locations of $\mathbf{x+1, x+2, \dots, x+2^{s-1}}$. The nodes of the form $\mathbf{[x+2^q]}$ are marked on the ring. The green lines represents links from the node x to other nodes.

Follow the picture and make sure you understand what $\mathbf{[x+2^q]}$ means - It is the "first" (clockwise) node with a name bigger than the number $\mathbf{x+2^q}$ on the ring.

This idea of wiring is also known as a Skip list.

New Search Algorithm

Let's describe the searching process with the new links structure. Assume that node x ($x \in \mathbf{B_s}$ is the name of the node) wants to reach node y. Node x will first check his own list of links, and see if he is already connected directly to y. If this is the case, x can reach y.

But x will not be that lucky every time. if y is not in x's links list, then x will choose

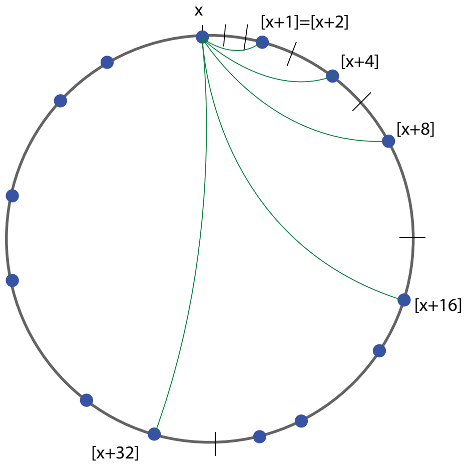


Figure 4: Logaritmic wiring.

the "closest" option - a node x_1 that is the closest x knows to y. By "closest" we mean the closest when walking clockwise. (As an example, the node just before x on the ring is the farest node from x).

x will ask x_1 if he knows y, and if he doesn't, x will ask x_1 what is the closest node to y known to x_1? Let that node be x_2.
x will keep going, until he eventually finds y. We should analyze this algorithm to make sure that indeed x eventually finds y, and also how many iterations it takes to find y.

Analysis

Let us start with the simple things. How many links every node has to maintain? By the definitions of links earlier, we know that not more than s links. We said that the size of the set **B_s** must be more than n, therefore $2_s \geq n$, which means s≥**log(n)**. Therefore every node maintains about **log(n)** links. This is generally a reasonable number, even for very large n-s.
Next, we want to know how long does it take for a node x to find some random node y. In fact, we want to be sure that x always manages to find y eventually.

If you are not in a mood for some math symbols, I give here a short description of what is going to happen. We are soon going to find out that in every stage of the search algorithm we get twice as close to y. As the size of the set **B_s** is 2_s, we are going to have no more than s stages before we find y. This also proves that we always manage to find y.

Now let's do some math. We define the distance (going clockwise) between two nodes a and b to be **d(a,b)**. If **b>a** then **d(a,b)=b−a**. Otherwise **$d(a,b)=2^s+b-a$**. (Think why).
Back to the searching algorithm, we can note that at every stage we are at point xt on the ring, and we want to reach y. We will pay attention to the amount **$d(x_t,y)$** at any stage of the algorithm.
We begin from x. If x is not directly connected to y, then x finds the closest direct link he has to y. Let that node be x_1. As x is linked to:
[x+1],[x+2],[x+4]...,[x+2s−1]
we conclude that:
d(x1,y)<d(x,y)/2
Let me explain it in a more detailed fashion: Assume that **y=x+q** for some q (The addition of **x+q** might be modulo the set **B_s**). There is some integer number r such that **2r≤q<2r+1**. (You could understand it by counting the amount of bits in the binary representation of q for example).

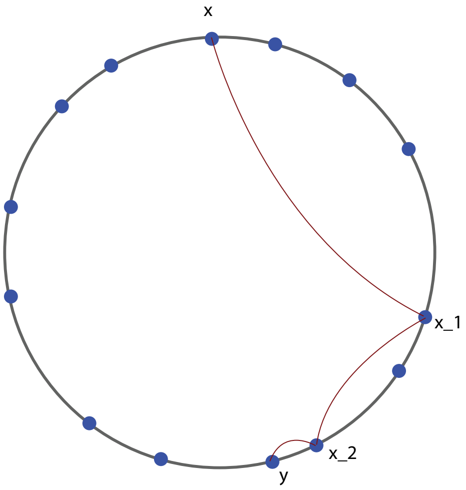


Figure 5: Illustrated search process.

Therefore the closest link from x to y would be **$[x+2^r]$=x1**. And indeed, we get that:
$d(x_1,y) = d(x_1,x+q) \leq$
$d(x+2^r,x+q) \leq q-2^r < q/2 = d(x,y)/2$.
So we get that **d(x1,y) < d(x,y)2**.
The same is true at the next stages of the algorithm (When finding **$x_2,x_3,\ldots$**, therefore we conclude that on every stage we get twice closer to y, compared to the previous stage. Finally we get that:
$d(x_q,y) < d(x_{q-1},y)/2 < d(x_{q-2},y)/4 <\ldots<$
d(x,y)/2q
We know that the initial distance **d(x,y)** is no more than 2^s, therefore in at most s stages we will reach distance 0, which means we have found y.

If you are a careful reader, you might be worried at this point that s might be much more than **log(n)**. This is in fact true. It is also true that in some worst case scenarios the amount of stages for the search algorithm will actually be **s**, even if **log(n)** is much smaller.
However if the names of the nodes are chosen somehow uniformly from the set Bs, we should expect better results which are much closer to **log(n)**.

Some words about Chord

Congratulations, you now know how to wire a collection of n nodes so that they can contact each other quickly, and at the same time each node doesn't have to remember too many addresses of other nodes.
The construct we have described is related to an idea called The Chord DHT. You can find the original article here

Distributed Hash Tables (DHTs)

Lets discuss an important use case for the structure we have found so far. We want to be able to store a large table of keys and values over a large set of computers. This is usually called a Distributed Hash Table (DHT).

The main operations that we want to be able to perform are as follows:

- **set_value(key,value)** - Sets the value of "key" in the table to be "value".
- **get_value(key)** - Reads the value of "key" from the table.

The cool part is that we can invoke those operations from any of the computers, as all the computers have a symmetric role in the network. Instead of letting just one computer deal with requests from client, theoretically we could use all the computers on the network. (Though we might have to deal with some synchronization stuff, which are outside the scope of this document).

There are still some questions to be asked here. What kind of values can the keys be? Must they be numbers, or could they be something else? Maybe strings?

Lets begin with the case in which keys are also from the set $\mathbf{B_s}$. This is not always very realistic, but it would be easier to solve at this point. In that case, the keys are in the same "space" as the names of nodes.

We could let node **[k]** keep the value of key k, where **[k]** is the "last" node (clockwise) that has a name not bigger than the number k.

To invoke **set_value(key=k,value=v)**, we first search (Using our search algorithm) for the node that is responsible for keeping the key k. This is done by searching for the value k. We are going to find the node **z=[k]**, which is exactly the node that has the responsibility to keep the key k. Then we just ask the node z to update k to have the value v.

To invoke **get_value(key=k)**, again we search for k, and find the node **z=[k]**. We then ask z what is the value that corresponds to the key k. z will then tell us the value v.

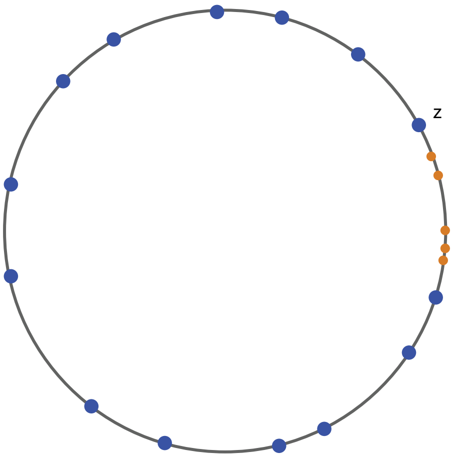


Figure 6: The node z (A blue dot), and some keys that z is responsible to keep (Small orange dots). The keys and node names are of the same kind (Both are from Bs, so we can also draw them on the ring according to their value. The next node (clockwise) after z marks the end of the domain z has responsibility over.

Dealing with complex keys

But what if our keys are not from the set Bs? Maybe the keys are strings? Maybe they are names of files, or people? In that case all we need is some function $\mathbf{f:K \to B_s}$, where K is the world of keys. Hopefully the function f will also be some kind of a random function, which means a few things:

- It is very unlikely for two keys $\mathbf{k_1,k_2}$ to satisfy **f(k1)=f(k2)**. (A property also known as Collision Resistance).
- The keys will map evenly as possible between all the elements inside the set $\mathbf{B_s}$. We don't want to have too much load of a few of the computers.

If you were wondering where you can get such a function, don't worry. We have a few of those functions. They are called Cryptographic Hash Functions.

Now that we have the function f, we will define two operations:

- **set_key_generic(key=k,value=v)** will invoke **set_key(key=f(k),value=v)**.
- **get_key_generic(key=k,value=v)** will invoke **get_key(key=f(k))**

And we get a DHT for a generic key space.

Final Notes

We have introduced a special way to wire a set of computers, so that we don't use too many wires, and at the same time it is easy to find any computer quickly. A major use case of this construct is the idea of DHT.

Our main construction follows the idea of the The Chord DHT, however there are other possible designs for DHT which we haven't talked about. Our space of names was a ring, with a distance function of walking clockwise. There are other spaces with different distance functions that give nice results. One notable example is the Kademlia DHT, which uses XOR as a metric.

We discussed the problem generally, but we didn't address a few important issues. We didn't address stability issues (What happens if some node on the way goes offline just when we want to search for some key?) and security issues. (What happens if a node gives us a wrong value for the key? Could an adversary block users from getting the value of a specific key in the DHT?)

We will think about those topics and how to deal with some of them in the next articles.

http://www.freedomlayer.org/articles/
dht_intro.html

Stratos II

Reaching beyond

Kevin Kouwenhoven

During the last years Delft Aerospace Rocket Engineering (DARE) designed and realised the Stratos 2 project: A student build sounding rocket that will carry an on-boardscientific payload to an altitude of 50 kilometres. The Stratos II rocket is a 7 meter long single stage hybrid rocket. It runs on nitrous oxide as oxidizer and a mixture of paraffin, sorbitol and aluminium as fuel. The engine, dubbed DHX-200 Aurora , has a total impulse of 200 kNs spread over a total burn time of around 30 seconds.

After the successful launch of the Stratos I rocket in 2009, the dream was born to go much higher and eventually break the boundary of space. 50 kilometers altitude was considered to be the next milestone goal towards space as it involved more complex systems but was technically within reach of the society. In 2012 Dutch Space gave its support to the project when it became the main sponsor. This marked the start of the building and testing phase of the Stratos II project. On the 12th of June 2014 DARE unveiled the Stratos II rocket to the public. After this DARE performed multiple dress rehearsals to practice and test the protocols needed for the launch. This included for example actuation tests but also the procedure of erecting the launch tower. During the weekend of 13 September the Stratos II rocket was packed in the custom made transport boxes and shipped to Spain.

On Tuesday the 23th of September the first of three groups arrived in Sevilla, Spain, marking the start of the launch campaign. At this moment the launch was scheduled for the 1st of October meaning that we had 1 week to prepare and integrate all subsystems and payloads. During this week the Stratos II launch crew worked in one of the workshops at the INTA base in Spain. The subsystems were unpacked, checked and assembled using precise check lists. The weekend marked the arrival of the Nijmegen payload, an interferometry measurement system, and the crew of the Radboud University Nijmegen responsible for the payload. On Monday and Tuesday the payload were integrated and placed into the payload compartment of the Stratos II rocket.

On Tuesday the 30th of September the Stratos II management decided to postpone the launch to Thursday the 2nd of October because of unfavourable weather conditions. On the 2nd of October all systems were ready for launch. The Stratos II rocket was mounted to the Launch tower at the correct elevation. After the results of the weather balloons came in the filling procedure was started. During the detachment of the oxidizer feedlines there was a visible leak in the oxidizer tank visible. Normally this is not a problem because small leaks are taken into account during the design of a propulsion system. For example during the launch of the Ariana rocket it is also visible that some oxidizer leaks out of the tanks. Parallel to the small amounts of oxidizer leaking the Flight Termination System lost signal with the base. Losing this signal means that the main safety system doesn't work which means we cannot launch the rocket. Because the countdown was paused due to the issues with the flight termination system the Stratos II rocket kept on leaking more and more oxidizer which meant that the oxidizer tank started to depressurize. During the

> Parallel to the small amounts of oxidizer leaking the Flight Termination System lost signal with the base.

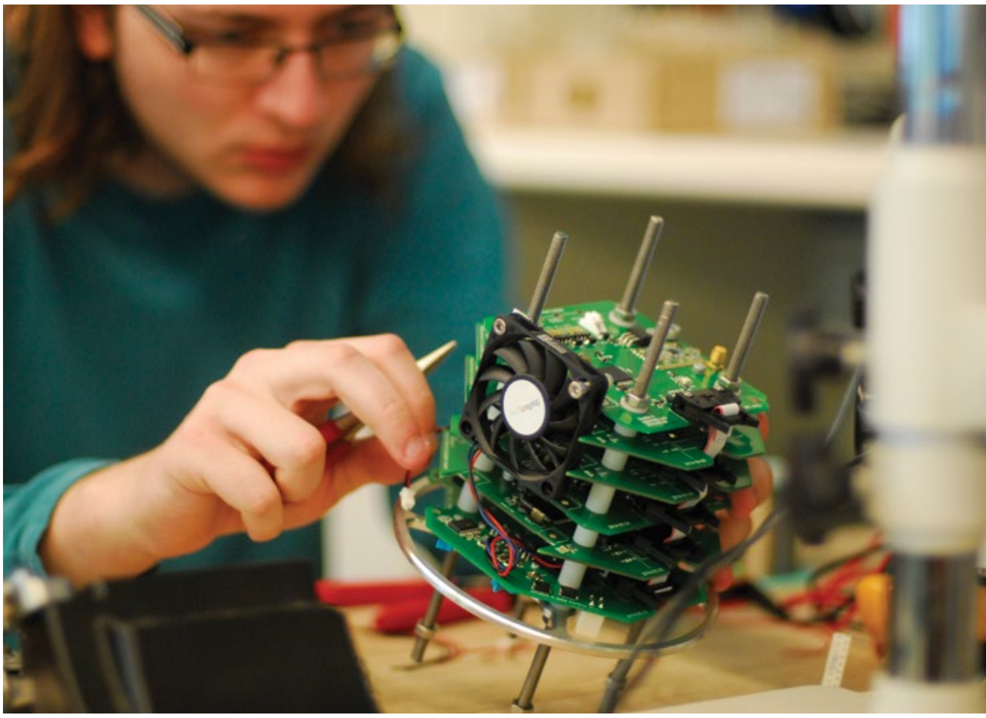
paused countdown the oxidizer in the tank was reduced form 75 kg to 40 kg. Because the feedline was already cut we couldn't repressurize the oxidizer tank. This in combination with the limited amount left in the launch window led to the decision to abort the launch and reschedule it to Friday.

On the final launch day (October 3rd), after the rocket was assembled and mounted onto the tower, another set of tests was performed on the Flight Termination System (FTS). It was found that the system did not work. After many stressful moments with the fear that the last launch opportunity could be missed, the cause of the error was located. It was found that the resistors in the negative feedback amplifier of the FTS were mixed up. This happened after the repairs that took place after the launch abort the day before. The wrong resistors resulted in a high gain in the amplifier. This again caused the signal received to be amplified significantly such that the signal clipped. After the problem was found the FTS was taken out of the rocket, resistors were soldered in the correct configuration and then reintegrated again.

At this point we were ready for the last and final launch window. During the launchwindows the airspace and sea around the base are closed for all traffic. To check if the sea is really cleared of boats 4 ships belonging to INTA scout the closed off area and redirect any ships entering the area. During the launch different stations are manned by DARE and INTA personnel. One of these stations is the control room. The control room initiates the countdown and makes sure that all other stations are synched to this countdown. The communication between these systems is provided by radio. One of the other stations is the hangar nearby the launchpad which contains the actual launch button. All personnel not manning one of the stations resides in the conference room. In this room the different video sources are visible, including the radar and optronics system, making sure everyone can watch

Electrical engineering within DARE

Electrical engineering is an important part of DARE. Rockets are mainly autonomous robots. To create these autonomous systems we design and build our own electronic systems. This ranges from simple counters for small rockets to complex systems using both microcontrollers and FPGA's. Because of this we are part of the robotic institute of the TU Delft and do we have our own electronics lab in the EWI faculty. The work of an electrical engineer within DARE varies a lot. For example: we design measurement electronics for engine testbenches, we create complex flight systems which integrate different payloads and we design Telemetry systems. This work is done under the banner of the electronics team of DARE.



the launch.

At half past twelve on Friday, the final launchday, we got the final confirmation that we had a "go" for launch at 13:30. The leak from thursday was avoided by cutting the feedline 10 seconds, instead of the normal 30 minutes, before ignition. During the filling process we found a new leak in the oxidizer tank. This leak developed because we had to assemble and disassemble the feed system mutiple times to test the Flight Termination System. This means that the stratos II Rocket was again leaking oxidizer. During the countdown we again found that the Flight Termination System didn't receive the signals from the base. This meant that the countdown had to be paused which caused the danger of tank depresurizing. During the pause it was decided to test the FTS one more time with the Intermediate Frequency Amplitude (IFA) of the signal increased. With the IFA increased the system regained and maintained signal which meant that we had a "go" for launch. Because the Stratos II rocket had already leaked some oxidizer during this pause the rest of the procedures where shortened so that we could launch as fast as possible. The range safety officer started synching the faster countdown with all the control stations. With only a few seconds to go until launch the whole team started to count down together! 5, 4, 3, 2, 1, 0...

> " We chose these challenges not because they are easy, but because they are hard."

But no launch happened. It was immediately visible that the engine had suffered from a misfire. The launch was immediately aborted to ensure safety. This meant that we had lost our final opportunity to launch.

When the oxidizer tank was depressurized and the launchpad declared safe we started investigating what went wrong. It became clear that the main engine valve did not open. This means that there is no fuel flow through the combustion chamber which caused the misfire. The reason why the engine valve didn't open was that the actuator was frozen in place due to a nitrous oxide leak.

Back at the camping, and recovered from the disappointment, all the members decided that this will not mean the end of Stratos II. We will revise the system and we will launch the Stratos II rocket. To quote the Stratos 2 teamleader, who in turn quoted JFK: " we chose these challenges not because they are easy, but because they are hard.".■

OGD ict-diensten
samen slimmer

Ben jij (m/v) op zoek naar de ideale bijbaan?

En handig met computers?

Dan hebben wij voor jou:

- Leuk werk bij diverse organisaties
- Doorgroeimogelijkheden
- Gezellige borrels en leuke uitjes
- Gratis cursussen en trainingen
- Een goed salaris

> **www.ogd.nl/werken**

Career Column

Stephen van 't Hof

Working on a master thesis is very time consuming. However, do not postpone your job search. You will get your deserved rest after graduation and then the question will appear: "What next?" After my studies in Delft I have started working as a Junior Consultant at Lloyd's Register Rail Europe.

To be honest, finding a job was not as easy as yelling "I'm done studying!" and waiting for the magic to happen. When I started working on my master thesis, I also started investigating the job market. As a biomedical engineering student, I was looking for a job in the medical devices industry. Obviously I failed on that part, as I am now working in the rail industry. For my application at a well known medical devices company, I had to finish an online test and visit Eindhoven, Drachten and finally Tilburg. The week after they told me I had finished second and thus did not get the job. Bummer. Then I found some medical start-up in Leiden, but they could not offer me a pay. With no ways to finance my existence I could not accept this job.

It was time to turn the tables. What companies where looking for me? I subscribed myself to temporary employment agencies and visited the Techniek-Bedrijvendagen, De Delftse Bedrijvendagen and EEMCS recruitment days. There I found a lot of companies looking for electrical engineers. Especially electrical power engineering and embedded system programming jobs were being thrown at me, even though my master studies did not fit. After visiting some in-house days and job interviews I decided I want to work as a technical consultant, preferably in the Randstad.

Lloyd's Register Rail Europe had a stand on De Delftse Bedrijvendagen with some very enthusiastic employees. They convinced me to visit their office on an in-house day. The experience was great, the employees were friendly and the case study was interesting. Therefore I applied for the job of Junior Consultant, which I am now.

As a Junior Consultant in the rail industry, I get very diverse tasks. I have worked on a sensor system for measuring temperatures, I have tested a new train on the track, I have instructed NS personnel and I have made a literature survey about train detection. The one day I might be installing a measurement system on a train, the next day I might do a presentation for a client. Diversity is what makes my work fun! Utrecht is a great city and our office is located right above the central station, next to the city centre. Although I did not find a job in the medical devices industry, I am really enjoying my work. ■

Discovery Study Tour

Het verslag van de grote studiereis

Sjoerd Bosma

Maandag 18 augustus 2014: dag 1 van de Discovery Study Tour, een reis van epische proporties georganiseerd door de Reiscommissie van de ETV. Een dag die voor 24 van ons het begin van een drieweekse reis door de Verenigde Staten markeerde. Al maanden was de commissie bezig met het bellen, mailen, Skypen, Whatsappen en sms'en van bedrijven en universiteiten in de VS. Ook de deelnemers waren al weken met hun hoofd niet meer bij de studie maar bezig met cases om de financiën op orde te krijgen. Maar dat lag het moment dat we verzamelden voor de trein naar Schiphol al een eeuwigheid achter ons, leek het. In de vroege morgen nog snel een versnapering in de vorm van Vlek, vergezeld door een afscheidsspeech door Ludo en we waren good to go.

Het doel van de reis was drieledig: in eerste instantie was het voor de reizigers bij uitstek een kans om Amerika (beter) te leren kennen. De gigantische diversiteit die het land rijk is, is ons niet voorbij gegaan. Ten tweede was het een mooie kans om de bedrijfscultuur te proeven. Velen hebben immers in Nederland al enige ervaring in de vorm van stages of BEPs opgedaan bij bedrijven, maar hoe doen ze dat aan de andere kant van de oceaan eigenlijk? Ten derde was het niet alleen een zakelijke reis, maar ook een culturele. Alle hoogstandjes van cultuur in de regio's die we bezochten zijn langsgekomen.

> *"De eerste week stond in het teken van de East Coast "*

De eerste week stond in het teken van de East Coast, Boston en New York specifiek. De eerste stop was MIT op de voet gevolgd door Harvard, twee van de meest prestigieuze universiteiten uit de stad (technisch gezien liggen ze echter in Cambridge, MA). Met geweldig indrukwekkende campussen, leuke gidsen en professoren schuwden ze niet een machtsvertoon van academica voor ons neer te zetten.

In Boston verbleven we in een comfortabel hostel met prima ontbijt en goede verbindingen naar de binnenstad. Overigens was dit een terugkerend thema, ook het hostel in New York was goed uit te houden. Het hotel in Houston was dusdanig luxe dat we vermoeiende week

aan de oostkust heerlijk achter ons konden laten en de ochtenden fris en fruitig weer konden beginnen. Een hostel in Austin was op kruipafstand van de bekende Sixth Street met genoeg uitgaansgelegenheden voor de complete TU en in Dallas was bij het hotel een patio met BBQ mogelijkheid! In San Francisco waren we eerst bij een idyllisch hotel waar je jezelf een Koning waande en de laatste overnachtingsplaats in het centrum van Frisco had fantastisch uitzicht over de baai en Alcatraz. Oh, en had ik al gezegd dat er bijna overal een zwembad en hottub waren? Enfin.

"De reis vervolgde zich dus in zuidwaartse richting: Texas"

Goed, zo ver waren we nog niet in het verhaal. In Boston stond nog Schlumberger op het programma, een bedrijf dat metingen verricht bij olieboringen, en als afsluitertje het Boston Museum of Science voor sommigen, voor anderen een historische wandeling die de Patriot in ons naar boven bracht. In New York Columbia University, een universiteit van het hoogste kaliber dat geweldig mooi in Manhattan gesitueerd is, Philips Medical Research en IBM met een bijzonder mooi gebouw (protip: bezoek de Wikipedia pagina van Thomas J. Watson Research Center voor epische panorama).

Al in de eerste week werd een groot verschil tussen Amerika en Nederland kenbaar: Amerikaanse werkplekken hebben geen ramen. Een tekenend voorbeeld is daarbij bovengenoemd IBM gebouw dat een gigantische gang heeft die geheel van glas is, maar de kantoren zitten aan die gang of nog dieper in het gebouw zonder mogelijkheid naar buiten te kijken. Bij Schlumberger en Philips kregen we hetzelfde beeld: leuke gebouwen met veel glas naar buiten, maar de werkplekken zitten niet bij de ramen. In de eerste week was het gebruik van cubicles niet opgevallen. Dat zou kunnen komen omdat ze wat Europeser zijn, maar misschien was het ook wel toeval. In de twee weken die volgden zagen we bij ieder bedrijf wel cubicles: grijze afscheidingswanden tussen werknemers geplaatst in een grote ruimte (zonder ramen, uiteraard). Navraag leerde dat dit in Amerika heel gebruikelijk is en dat alleen hooggeplaatsten een eigen kantoor hebben. Wel was er nog enige variatie in de hoogte van de muurtjes: bij sommige bedrijven kon je er staand overheen kijken en bij anderen waren het echt metershoge installaties.

De reis vervolgde zich dus in zuidwaartse richting: Texas stond op het programma. Houston, Austin en Dallas stonden in die volgorde op het programma. Fundamentele natuurkunde bij het Texas Center for Superconductivity, smart grids bij zowel CenterPoint als Pecan Street, telecommunicatie bij Alcatel-Lucent en analoge elektronica bij het Texas Analog Center of Excellence waren leerzaam en interessant voor een zeer breed publiek aan elektrobazen. Maar het was natuurlijk veel meer dan dat: integratie met lokale studenten hadden we in Boston

al beleefd maar kregen we nu nog een keer in Texas. Overheerlijke lunch, gezellige babbels en leuke presentaties van locals maakte ook Texas voor ons onvergetelijk. National Instruments (NI) en Texas Instruments (TI) zijn misschien wel de twee bekendste bedrijven uit het lijstje die we in het zuiden bezochten en ook daar zijn we hartelijk ontvangen.

Tijdens de reis was de commissie natuurlijk nog hartstikke druk bezig met bevestigen van afspraken, ophalen van busjes en andere praktische zaken maar ook de deelnemers werden van de straat gehouden. Zo was er een fotocommissie die met professionele apparatuur en dito ervaring de mooiste foto's gemaakt heeft, een voorverslag commissie die voor het aanvangen van de reis al klaar was met een boek over de reis met interessante feitjes en wetenswaardigheden over de te bezoeken bedrijven en steden en een dagboek commissie (bravo!) die dagelijks een nieuw artikel over het reilen en zeilen van de groep op het blog geplaatst heeft. Ten slotte is de eindverslag commissie die na afloop van de reis bezig is om het verhaal helemaal af te ronden met een mooi boekwerk dat de reis nog even samenvat.

Terug naar de reis zelf: Silicon Valley stond namelijk op het programma! Voor iedereen sowieso een droom om daar eens geweest te zijn, maar om daar van binnenuit een sneak peak te krijgen van zulke grote bedrijven is echt waanzinnig. Synopsys, bekend van Electronic Design Automation tools, Synaptics van de touchscreens, Palo Alto Networks van de hardwarematige firewalls en Google, bekend van nouja.. alles, stonden op de to-do lijst. Ook topuniversiteiten UC Berkeley en Stanford ontbraken niet. Power Engineering is gezien bij PG&E Research (een soort Nuon maar dan hipper) en Trans Bay Cable, een operator van een gigantische onderzee DC kabel door de baai van San Francisco.

"een sneak peak te krijgen van zulke grote bedrijven is echt waanzinnig"

Al met al kan ik denk ik voor alle reisdeelnemers spreken door te zeggen dat deze drie weken aan het eind van de zomervakantie echt onvergetelijk waren. We zijn zo warm ontvangen door ieder bedrijf dat we bezochten, zo hartelijk gegroet door alle studenten die we daar tegen gekomen zijn en hebben zulke mooie natuur gezien dat we – ondanks de verloren hersencellen door alcohol consumptie – kunnen terugkijken op een tijd waarin we onszelf écht verrijkt hebben om een zeer aangename manier.

Natuurlijk is dit stuk veel te kort, ik heb het nog niet gehad over de pubcrawls, sportsbars, ice bucket challenges, prof diners, Six Flags, fastfood ketens, busje 1, spetterende raamsessies, Delfsch Blaauwe Bordjes, shooting ranges en nog talloze ervaringen die we hebben opgedaan tijdens deze reis. Voor een verzameling van alle dagboek stukjes zou ik je dan ook willen doorverwijzen naar de site, http://reis.etv.tudelft.nl, waar ze allemaal nog eens rustig na te lezen zijn. Ook de foto's kun je daar vinden en een overzicht van alle bedrijven die we bezocht hebben. Voor de wat meer obscure verhalen moet je natuurlijk een van de reisdeelnemers aanspreken, ook de deelnemerslijst vind je op de site.

Tot slot wil ik de reiscommissie: Tim Feenstra, Benjamin Gardiner, Wieger IJtema en Erik Wouters hartelijk bedanken voor hun inzet afgelopen jaar en de komende tijd gedurende de nasleep. Zonder jullie was dit hele avontuur nooit verder gekomen dan de /Pub. Geniet nog even van jullie T-shirts! Ook de professoren die ons op verschillende momenten tijdens de reis hebben vergezeld, bedankt. Het was erg gezellig om jullie beter te leren kennen en bijzonder leerzaam en inzichtelijk! Voor iedereen die twijfelt om mee te gaan op een studiereis: Ga zeker mee! Voor mij is het de meest indrukwekkende kennismaking geweest met de elektro cultuur in het buitenland en ook al kende ik het gros van de deelnemers voor de reis niet: aan het eind heb ik er 23 (nieuwe) vrienden aan over gehouden. ■

An introduction to Botnets

Largest hack in history

Advertorial by Pinewood

A few weeks ago, one of the largest hacks in the history of the Internet was made public. A security company in the U.S. discovered an enormous collection of records containing confidential information gathered from thousands websites. These records were collected by Russian hackers and contained thousands of username-password combinations. Their primary purpose was sending spam, but these records could have been used for many more purposes. The hackers were able to capture this enormous amount of credentials by using botnets. These botnets enabled the hackers to automate the scanning for vulnerable websites, so they could later hack the website and collect the data. In this short article, we will give a brief introduction on botnets.

What are botnet, and where do botnets come from....

In technical terms, a botnet consists of network (net) of 'robots' (bots) that communicate in order to perform certain tasks. Using an encrypted channel, the bot herder (or bot master) remotely controls the botnet by sending tasks to a Command & Control (C2) server, which in turn distributes the commands to the bots. The formation of a (malicious) botnet is traditionally staged by using malware-infected computers. When the computer is infected, e.g. through an infected e-mail attachment or a malicious download from a website, the malware connects to the C2 server, making it part of a botnet. This usually happens without any knowing of the owner of the computer. Now, the infected computer (zombie) is controlled by the bot herder and does as he or she bids.

Historically, botnets were mainly used for legitimate purposes, such as controlling chat channels on Internet Relay Chat (IRC). Nowadays, botnets have acquired a largely negative connotation, since they are often employed for malicious activities. The earliest appearance of malicious botnets goes back to 1999, when malware was spread that introduced the concept of connecting to an IRC channel to listen to commands. These early versions were able to execute scripts on the hosts in response to IRC events. Addresses of C2 servers were hardcoded, making them vulnerable to eradication in case the C2 communication was intercepted or the C2 servers were taken down. Later, these vulnerabilities were mitigated and more resilient botnets appeared.

The earliest appearance of malicious botnets goes back to 1999

Nowadays, botnets can take several forms. In 2003 the first manifestation of a peer-to-peer (P2P) botnet was discovered. In a P2P botnet every peer in the botnet can act as a C2 server, while none of them really are one. Each bot keeps a list of other bots and exchanges commands, configuration files and executables with its peers. This essentially solves the point of failure that designated C2 servers represent. Another common form that is seen is where botnets use social networks as a C2 platform. Hackers prepare a user account and upload encrypted commands to this account; the bots read from the account and follow the instructions. Twitter, Facebook and Evernote are examples of social networks that have been successfully employed in this way; and since social networks are so popular, this C2 traffic is very hard to uncover.

Uses of botnets

The architecture of a botnet is often designed with its purpose in mind. Notable uses of botnets include:

1. **Distributed Denial-of-Service attacks (DDoS).** DDoS attacks are for rent and can be as cheap as $100,- per hour. This "malware as a service"-botnet overloads the victim with traffic, effectively depleting all its resources. The result is that its (web) services become unavailable for legitimate users, leading to financial and reputational losses. In 2013, several

large financial institutions in the Netherlands (but also worldwide) were victims of DDoS attacks.

2. **Stealing information.** In 2013, the Pobelka botnet was uncovered. This botnet contained an enormous amount of information on critical infrastructures and other sensitive objects. Botnets are designed to steal sensitive information and intellectual property. This information can then be sold on the black market or used by state-sponsored entities for strategic purposes.

3. **SPAM.** Spam is the considered one of the oldest (and still most common) uses of botnets. Botnets are instructed to send billions of emails daily. In 2008, the Srizbi botnet was taken down, consisting of 500,000 infected nodes. This botnet was reported to be capable of sending around 60 billion spam messages a day.

> The capabilities of botnets that use mobile devices are far more advanced than their desktop counterparts.

Experts estimate that about 85 percent of all email traffic is spam.

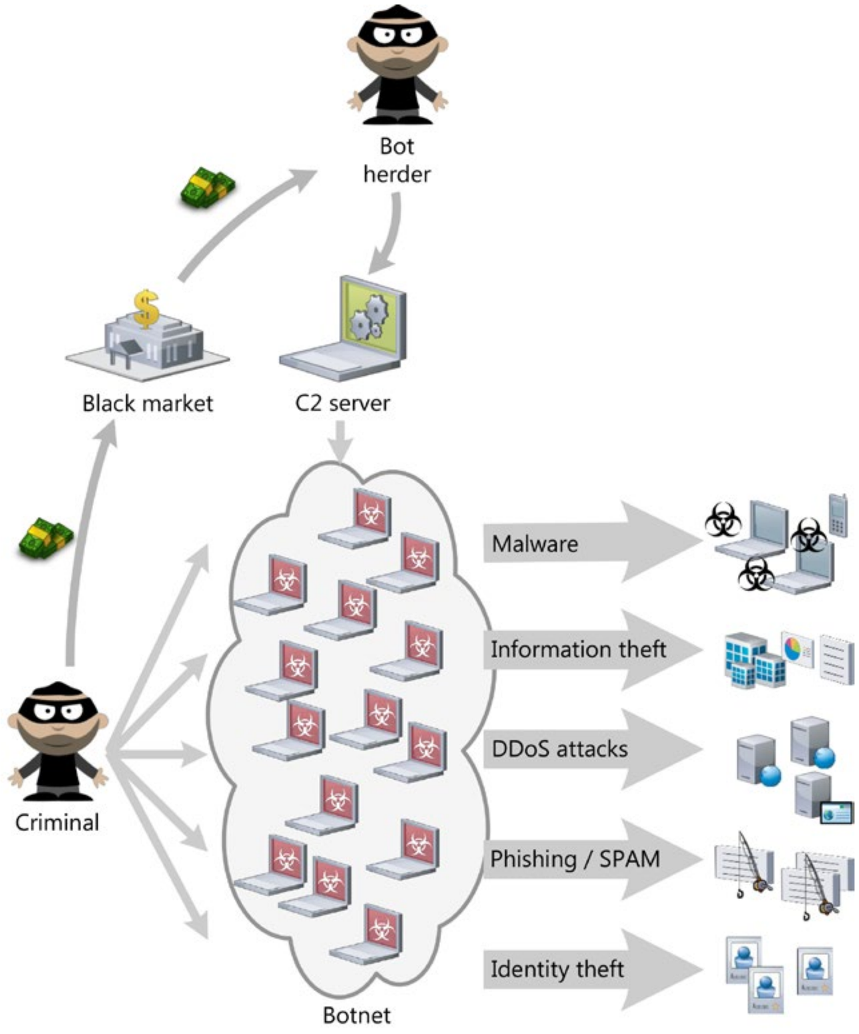
4. **Click Fraud.** Botnets can be used to gain financial advantages through popular advertisement add-ons. The hacker sets up a fake website (or automates this and sets up thousands of websites) with some advertisements and makes an arrangement with some hosting company that pays for clicks, e.g. Google's AdSense. By letting bots click on the advertisements, the hackers earns his money.

Experts estimate that 40% of all clicks on advertisements come from bots. An analysis on the ZeroAccess botnet, with up to 1.9 million machines, reported that it clicks 140 million ads per day. The analysis estimates that, to legitimate online advertisers, this botnet costs USD 900,000 of daily revenues.

5. **Identity Theft.** Bots can be used to steal personal information from the infected computer. Through key loggers or traffic sniffers, bots can steal user credentials and other personal information. With this information hackers are able to rob back accounts and other financial services, such as a Paypal.

Advancing technologies...

With the introduction of smartphones and tablets, hackers have gained a new playground for their activities. Security researchers have discovered a huge botnet running of smartphones of mobile users in China. These mobile botnets are formed using malware-infected applications or bad SMS messaging. The capabilities of botnets that use mobile devices are far more advanced than their desktop counterparts. Firstly, smartphones offer access to unique functions, such as location services, telephony services or contactless payment services (NFC or Near Field Communication), which increases the attack surface. Secondly, due to the



A flowchart to demonstrate the principle of malicious botnets.

highly personalized use of smartphones, these devices contain large amounts of personal information and functions, such as contact lists, agendas, photo's, banking applications, two-factor authentication applications (soft tokens), interactive messaging application (WhatsApp, Telegram, iMessage), etc. This personal information can then be used for identity theft or illegal transactions and mobile banking attacks.

> Very recently, researchers in the U.S. built a botnet using (free) public cloud services ... it is also legal!

Very recently, researchers in the U.S. built a botnet using (free) public cloud services. Through "freemium" accounts, companies offer free cloud space to developers to host their applications. Often these companies resell resources owned by larger companies like Amazon and Rackspace. Since many of these service providers only require an e-mail address for identification, the registration procedure can easily be automated. This type of botnet is not only free; it is also legal! The botnet is built without trespassing any security measures or stealing user credentials. (As soon as the botnet is used for malicious activities, it becomes illegal.) The major threat that emerges from this type of botnet is the high availability and connectivity of the bots, making it a "weapon of mass destruction" for Distributed Denial of Service attacks. Moreover, since the attack is launched from "reputable" cloud providers, it will be hard to divert the attack without cutting off legitimate traffic.

Conclusions

Mitigation techniques against botnet require joint actions between law enforcement and private industry. In some cases tech-giants, like Microsoft, have already embarked on close cooperation with governments. Legislative regulations and penalties for those who develop, spread or employ botnets should be recognized globally as bot herders move their control organization to countries where governments are tolerant and law enforcement is weak.

> In some cases tech-giants, like Microsoft, have already embarked on close cooperation with governments

Technical countermeasures are becoming more difficult with the advancing technologies used by botnets. To stop the spread and infections of botnets, it is important to increase the general awareness of cyber threats and come with a set of best practices to contain the infection. We conclude this article with a (short) list of countermeasures that everybody should adhere to:

1. Always use (up-to-date) anti-virus and anti-spy software from a trusted source. Malware forms the foundation on which botnets are built. Anti-malware software detects malware on your system and helps removing it.

2. Always use up-to-date versions of software (both desktop and mobile). Hackers make use of (known) vulnerabilities in software. By keeping your software up-to-date, these vulnerabilities are minimized, making systems less vulnerable for hacks.

3. Always use official channels for software (official marketplace for Android and Apple App Store for iOS). Rogue software channels often "free", hacked applications. This hacked software is often infected with malware, infecting your (mobile) system

4. Always use strong passwords and preferably use a password vault to manage them. Strong passwords are hard to guess by people of machines and less vulnerable to attacks . People can only remember a small number of strong passwords. By using a vault, you can diversify passwords for various websites/services and keep passwords save. Examples of vaults are KeePass, 1Password and LastPass.

5. Be careful with opening attachments from suspicious e-mails. E-mails from an unknown sender, containing an attachment or link are always suspicious. Do not click on the attachment or link, unless you are sure the source is legitimate. Usually, these phishing mails contain malware and will infect your system, and make it part of a botnet. ■

Preethi Pradha Elavarasu on her career at ASML

Earning a living while learning on the job: "I am lucky!"

Preethi Pradha Elavarasu

It was her aunt pointing out the opportunities that made Preethi Pradha Elavarasu decide to follow a technical education in the Netherlands. So she flew in from her village in India and joined TU Delft as it 'had the highest ranking'. A few years later, she obtained her Master's in Electrical Engineering and found a job at ASML. Preethi: "It is a real big company where they produce very advanced machines. The nice thing is that here all kinds of technical disciplines are working together closely. I thought it the perfect environment for me. A company where I can learn a lot and er develop myself further."

Preethi started as an integrator in the hardware department. With her team, she added new blocks of electronics to modules of the machine and performed tests with it. Preethi: "When I started I was just a recently graduated student walking around in a very big company for the very first time. The good thing is that you're appointed a coach at ASML. During the first six months, you work closely together with your coach, so you can get acquainted with the company, the culture and, of course, the machines and their processes. This was really, really helpful . The machines at ASML are extremely complex. They consist of a lot of modules with an enormous amount of features and dependencies. It is difficult to get an overview and imagine how it all actually works."

IC enabler

For those who are not familiar with ASML, here is a brief explanation. ASML in Veldhoven manufactures lithography machines for the production of integrated circuits. The company supplies all major chip manufacturers in the world, including Intel, Samsung and tsmc - with products that defy the limits of what is scientifically possible. For instance, with the latest generation of machines, ASML is able to print lines and components on less than 20 nanometres. That is like printing a complete novel of 500 pages on one centimetre of a human hair!

Virtual Master

Meanwhile Preethi has found a new direction in the company. Preethi: "I initially chose electronics, but I found out that I could do more. For my Master's, I did a project at Bosch Security Systems on the wireless communication of audio and videoconferencing systems. Bosch wanted to switch from a non-standard wireless LAN protocol network to an open standard wireless communication protocol. The transfer met with several problems, among others interference. So I have setup a virtual network for analysis and performance testing of the wireless nodes: an all-software project. My coach at ASML noticed my competences and affinity in this field and he proposed a switch to the software department."

Smarter

"The first seven months in the software department, I did a test case automation. I wrote test scripts that automate a number of standard tests for new machines.

Now I work on system level in the Test Group and I am currently exploring automation there, too. My challenge is to re-use some of the tools that were used for test case automation. To use what we already know in a smarter way. Automation of testing can save ASML a lot of time, as each machine undergoes a load of standard test procedures before being shipped to the client. And in this project I can benefit from my experience in telecommunication. It is again about finding the link between simulated processes for automated testing. We use simulation for a large part because physical testing on the machine is too expensive."

> *"If you have the motivation and competences, everything is possible"*

Move made easy

"The switch to the software department was quite easy. ASML is really supportive to employees' career plans. If you have the motivation and competences, everything is possible. Furthermore, ASML invests in the personal development of its staff; I am, among others, following a Dutch course. And this is very useful for my Inburgeringscurses (newcomers' course)! However, the openness at ASML was also something I had to get used to. In India, people don't say what they think. Here, people you don't even know just say it if you're doing something wrong. This was quite shocking for me at first, but in the end, it works much better. If you're in development you need good feedback. I am very happy with my job. Developing new things is fun. And for the future? I am thinking of management. For starters the management of a project. It allows you to work with a lot of different people and teams, throughout the company. We'll see, so far I am lucky!" ■

The company

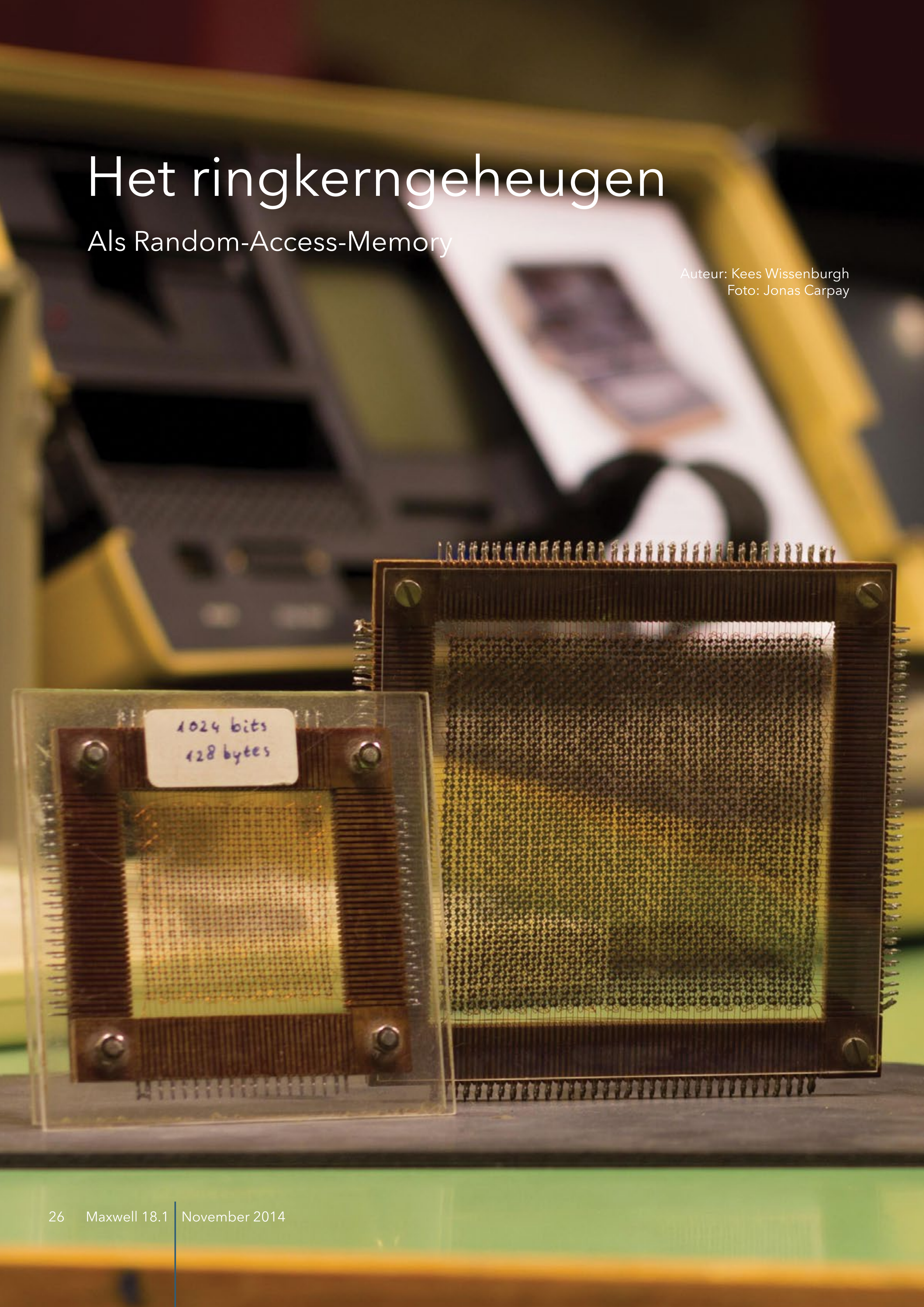
ASML is born from a joint venture by ASM International and Koninklijke Philips in 1984 and is listed on the stock market since 1995. In 1986 it brought its first system to the market, the PAS 2500. Nowadays the company is one of the most important suppliers of machinery for the semiconductor industry, especially steppers and scanners used for producing chips. Its headquarters are stationed in Veldhoven in the Netherlands. Research and development as well as assembly in cleanrooms happens there.

The biggest challenge for ASML to tackle is to keep producing machines that can create chips with smaller and smaller detail. The advantages for the customers are more memory or computing power while also consuming less energy. Shorter distances between components also improve performance. Now research is being done into using extreme ultraviolet light (EUV) which enable even smaller details on chips.

Het ringkerngeheugen

Als Random-Access-Memory

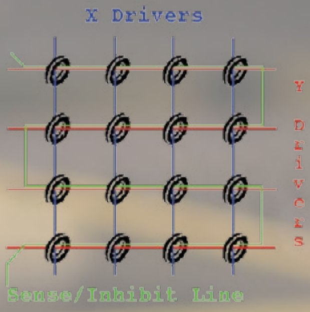
Auteur: Kees Wissenburgh
Foto: Jonas Carpay



De studieverzameling van de Faculteit EWI beoogt een beeld te geven van de ontwikkeling van het vakgebied. De begrippen geschiedenis en geheugen zijn nauw verbonden.

In de ontwikkeling van computers kwam al snel de behoefte om programma-instructies en data op een zelfde manier in te voeren en op te slaan in een geheugen; het werkgeheugen. Er werden in de loop der jaren binnen de toenmalige technische mogelijkheden diverse geheugensystemen ontwikkeld, maar gaandeweg werden de eisen opgevoerd. Zo moest de data na het uitschakelen van de computer niet verloren gaan (non volatile), verder moest elke geheugenplaats even snel bereikbaar zijn (random access) en wilde men het gebruik van mechanisch bewegende delen in verband met snelheid en slijtage vermijden.

Zo werd in 1953 het eerste zogenaamde ringkerngeheugen toegepast dat daarna gedurende ca. 25 jaar als werkgeheugen in computers heeft gefungeerd. Hoewel velen onderzoek verrichtten op dit gebied, was Jay Forrester (1918) van het MIT degene die het patent verwierf. Daar is trouwens nog jaren over getwist.



In 1964 betaalde IBM aan MIT nog het toen enorme bedrag van 13 miljoen dollar voor de rechten op het patent.

Het principe van het geheugen berust op het gebruik van de rechthoekige hysteresislus van ferriet, een ferromagnetisch keramisch materiaal. Een geheugencel bestaat uit een ring van ferriet, die 'linksom' of 'rechtsom' (digitaal 0 of 1) gemagnetiseerd kan worden, afhankelijk van de stroomrichting in een draad die door het gat van de ring is gestoken.

Een geheugensectie bestond uit een matrix van bijvoorbeeld 32 x 32 ringen. Om de geheugenplaatsen te kunnen adresseren werden door elke ring twee draden gestoken, X en Y lijnen. Alleen in de ring waardoor de X-lijn én de Y-lijn stroom in dezelfde richting voeren, wordt het magnetisch veld groot genoeg om de magnetisatie van de ring te kunnen veranderen. Om de in een ring opgeslagen data te kunnen lezen is een extra draad nodig (de sense lijn). Deze loopt seriegewijs door alle ringen. Aanvankelijk werd in verband met een betrouwbare werking ook nog een vierde draad door alle ringen aangebracht (de inhibit lijn). Later werden de sense en inhibit gecombineerd. De figuur toont het principe voor een 4x4 matrix.

Om het geheugen uit te lezen worden de geselecteerde X en Y lijnen zodanig aangestuurd dat het bit 0 wordt. Nu zijn er twee mogelijkheden:

Is het bit al 0, dan verandert er niets.

Is het bit 1, dan verandert de magnetische polarisatie van de ring. Die verandering induceert een spanningspuls in de sense lijn. Detectie van zo'n puls geeft aan dat het bit 1 was. Wás, want door het uitlezen is het bit veranderd van een 1 in een 0. Men zegt dat het uitlezen destructief is. Er moet daarom een schrijfcyclus volgen om het bit weer 1 te maken.

Schrijven gaat als volgt:

Om een 1 te schrijven moet stroomrichting in de corresponderende X en Y lijnen tegengesteld zijn aan die bij de leescyclus.

Moet er een 0 geschreven worden dan voorkomt men dat het een 1 wordt door een stroom door de inhibit lijn.

Een besturingscircuit zorgt er voor dat alles correct verloopt.

Zoals de foto toont bestaat een geheugensectie uit een vlechtwerk (matje) van ringetjes ter grootte van ca. 1 mm met door elk ringetje drie of vier dunne draden. Deze matjes werden handmatig in elkaar geregen, wat delicaat en inspannend werk was, dat onder een loep moest worden verricht en meestal werd toevertrouwd aan vrouwen.

In 1980 was de prijs van een 32kB ringkerngeheugen nog rond $3000.

Erik werkt als intaker op het OGD hoofdkantoor in Delft. We vroegen hem naar zijn ervaring met OGD als werkgever.

Hoe ben je bij OGD terecht gekomen?

Door een fulltime commissie op Virgiel ben ik een half jaar uitgelopen en moest ik een half jaar zien te overbruggen. Naast het trainen voor de Ringvaart wilde ik mijn tijd toch nuttig besteden en natuurlijk geld verdienen. Via-via hoorde ik van OGD en ik heb meteen gesolliciteerd.

Wat voor werk doe je nu?

Ik zou aan de slag gaan op een helpdesk. Samengevat komt dat erop neer dat je het aanspreekpunt bent voor de medewerkers van de klant als het gaat om ict- problemen. Concrete werkervaring of diepgaande kennis van ict heb je voor deze functie niet nodig, omdat je deze kennis opdoet tijdens het werk zelf of met behulp van de gratis cursussen die OGD aanbiedt. Bij OGD weet je echter nooit precies waar je terecht komt. Ik werd twee dagen na mijn intake opgebeld of ik het team van Werving & Selectie wilde komen versterken. Een week later kon ik beginnen als intaker!

Wat voor werkgever is OGD?

De sfeer bij OGD is informeel en gezellig. Elke week is er een vrijdagmiddagborrel waar je informeel met (nieuwe) collega’s biertjes kan gaan drinken. Daarnaast worden er veel activiteiten georganiseerd. Zo gaan we elk jaar op wintersport, zijn we met het team gaan wakeboarden, is er een zeilweekend, etcetera. Maar je merkt ook dat het een professionele organisatie is waar mensen gestimuleerd worden om zich te ontwikkelen.

Ben jij ook op zoek naar een uitdagende bijbaan bij een leuk en informeel bedrijf? Solliciteer dan direct!

>www.ogd.nl/werken

ETV Activities

What's happening?

Lunch Lecture TNO

by Jan de Jong

On Thursday September 29th the ETV organized the first lunch lecture of the new college year. As always we had the ETV supplying the audience with some delicious sandwiches to enjoy during the lunch lecture. The opportunity was given to TNO to present us with what they did as a company.

At first the presenter Edwin had a little talk about his life as an Electrical Engineer. He was a former TU Delft student and graduated in 2002. After this he moved to China and worked as a designer and project manager. He later joined TNO in 2006. He worked on several projects in different sectors of the company. He is now Technical Consultant / Projectmanager / Innovator at TNO Delft.

TNO itself is an independent research organization and consulting company. Their head office is in Delft and they work close with other companies to achieve new innovations and successfully complete projects in the technology sector. Their research is split in different sectors. Energy, Industry, Healthy living, Urbanization and Defense safety and security. The presenter explained what they did in all these sectors by giving an example of a project he worked on. An interesting project was called Smart Community Aruba. This was in the energy sector. Aruba wants to go to 100% renewable energy by 2020 and TNO helps them by building a small neighborhood, which is build for that. The main goal is to evaluate sustainable building techniques and efficiency measures and gain experience with the integration of renewable energy into the smart grid. After some questions from the audience being answered the presenter was thanked and the audience left.

Waterskiën

door Charlotte Treffers

Op woensdag 8 oktober is er een groep enthousiaste ETV'ers gaan waterskiën. Het was een regenachtige dag, maar dat mocht de pret niet drukken. We konden ons gelukkig in auto's verplaatsten richting de waterskibaan. Eenmaal aangekomen werd er snel omgekleed en na wat instructies kon het waterskiën beginnen. Voor vele personen was het op de waterski's staan een grote uitdaging en zo gebeurde het ook vaak dat men binnen een paar meter alweer naar de kant kon gaan zwemmen. Andere personen werden letterlijk uit de bocht geslingerd. Aan het eind van de middag waren er echter vele in staat een aantal rondjes op de waterski's te blijven staan. Na een uurtje waterskiën werd er onder het genot van warme chocomelk nog druk gesproken over deze coole activiteit. Deze sessie verplaatste zich later naar de /pub. Al met al was het een zeer geslaagde activiteit van de Zakcie.

Kwintjesavond

door Daniel Kappelle

Het mooiste weekend van het jaar, het Elektro Ontvangstweekend, ligt alweer een tijdje in het verleden. Toch was er onlangs een mooie gelegenheid om alle herinneringen op te halen: de kwintjesavond. Een avond met bier voor 20 cent, dat moet garant staan voor een mooie avond.

Op deze dinsdagavond druppelden steeds meer eerstejaars binnen nadat hun colleges waren afgelopen. Zij waren, volledig in de EOW stijl, gekleed in hun overalls en in hun linkerhand hadden zij een gevulde pul. Verder waren er genoeg anderen aanwezig, bestuurders, oud-bestuurders, mensen die op wat voor manier dan ook aan het EOW hebben bijgedragen en de EOW-commissie zelf.

Na een klein ontvangstpraatje konden dan ook de foto's van het EOW bezichtigd worden op een beamerscherm, wat hier en daar tot de nodige hilariteit leidde. Er werd genoten van het befaamde gerstenat en voor sommigen

een simpele doch afdoende maaltijd. Inmiddels eerstejaars integreerden met elkaar en met ouderejaars en de sfeer was alsof er nooit een einde aan het EOW was gekomen. Kortom een erg gezellige en geslaagde avond!

Eerstejaarsexcursie

door Philip van den Heuvel

Op onze, normaal vrije, vrijdagmiddag hebben wij als eerstejaars Elektro onze eerste excursie gehad. Vol verwachting kwamen we aan en deels met verassingen en enthousiasme gingen we weg. We hadden hiervoor een middag lang een leuk interessant en verscheiden programma gevolgd. Eerst kregen we uitgelegd wat YES!Delft was en wat haar doelen zijn, waarna het daarna steeds interactiever werd. We kregen leuke projecten te zien met interessante verhalen van de ondernemers en kregen een goed idee van wat ondernemen na je studie in zou gaan houden in de ondernemende en innovatieve omgeving van YES!Delft. Na allemaal voor een elektronische sleutel een doelgroep onderzocht te hebben werd de dag, net als elke vrijdagmiddag op YES!Delft, afgesloten met een gezellige borrel.

Board change

On Tuesday 9 September 2014 the 142nd Board of the ETV was discharged. The 143rd Board took their place.

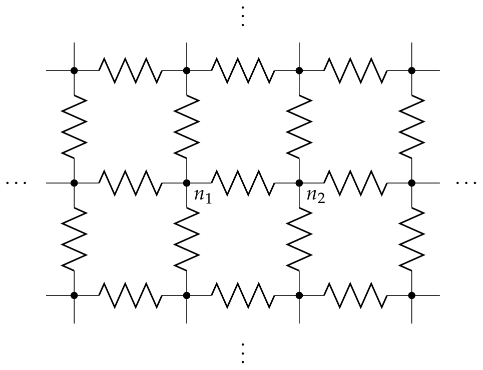
The Board consists of 5 people. Jonas Carpay is the President, Bart Kölling the Secretary, Sjors Nijhuis the Treasurer & Vice President, Attila Lengyel the Commissioner of External Affairs and Leon Loopik the Commissioner of Education.

They are fully committed to supporting all of the Electrical Engineering students in the coming year. This will be done by evaluating courses, organizing lectures, excursions, drinks and much more!

Puzzle

Starting this edition, this space will be reserved for a puzzle! If you have an interesting puzzle that's suitable for publication here, let us know at: maxwell-etv@tudelft.nl.

For this first edition, we have a classic electrical engineering problem. Imagine an infinite field of ideal 1 Ω resistors, such as the one in the schematic on the right. What is the equivalent resistance between two neighboring nodes (n_1 and n_2 in the schematic)? You can submit your answer by e-mail or by walking by the ETV board room.



Beleid
Techniek
Mens