UNIVERSITY OF COPENHAGEN

Organized Peer-to-Peer Network

Enabling interconnectivity and distributing workload

Mads Ynddal
SJT402

March 11, 2016

1 Research Problem

Is it possible to map out a local computer network, and from that, make selected devices available for outside access?

In the recent shift to cloud solutions, the focus has been on reducing the workload of the local administrators and reduce on-premise support. I this progress, it has been overseen, how we would solve scenarios, where devices on a local network, are inaccessible from the outside world.

A case could be a company with printers on a local network. These could be on an isolated network or on a local subnet. In the past, they had a specialized server on-premise to authenticate and proxy these inbound connections. The disadvantages to this, is the need to reconfigurate the network's forwarding and firewall rules. This require the company to have an expensive consultant or full-time employee to work on it. This can be challenging in the small/medium business market.

My solution proposition, is to install a small piece of software on the employee's machines. This will enable a form of peer-to-peer network, that will take care of interconnecting machines in a company – no matter if they are on-premise.

When interconnection has been accomplished, the software can also be used to distribute load away from the central servers. It will be at the core of the project, to minimize and distribute workload, to enable rapid scaling of the system.

1.1 Limitations

The software will not necessarily be evaluated in a real environment. It will, as much as possible, be tested on a virtualized environment of machines.

2 Tasks and planning

See attached PDF with Gantt-diagram.

2.1 Tasks

- **Protocol,**
  Before starting to implement the system, I'll outline a protocol and a system architecture. This is important for the further development, to keep on track and have a coherent system. As the development of the software progresses, the protocol might be revised to allow for improvements that might become apparent.

- **Language,**
  For the project, I will have to find and learn an appropiate programming language. Python, Rust and C# has all been considered, but I have chosen to take the challenge to learn the language Rust. Rust stands out for being compiled into native machine code and still allows for an almost automatic memory management.

- **Backbone,**
  When the protocol is outlined, the actual development can begin. This will focus around getting the infrastructure running between a few nodes.

- **Discovery,**
  The next step will be to give the nodes a form of logic about how to find other nodes. This will include a distributed hash table that will enable node to share information on the network.

- **Functionality,**
  This step will be to implement actual tasks for the node to carry out and make the network have a purpose. This will include file sharing and possibly an end-to-end TCP tunnel between nodes.

- **Encryption,**
  Through out the previous tasks, a dummy system will be used in-place of actual encryption. This step will be to find the tools to enable encryption on the system.

- **Advanced functionality,**
  This last step will be to develop a network-toolkit that nodes can use to check its environment. The tools will focus around checking firewalls and look for way to open a hole in it. This will possibly include hole-punching, UPnP forwarding and detecting blocked protocol and ports.