1 Introduction

- What is a Darknet, effects, features for its users
- Arising problems from its characteristics
- name the analyzing methods (and 1 sentence explanation)
  - analytic
  - simulation
  - emulation
  - test bed (and/or measurement in real world networks)
- Advantages of event based large scale simulation and in general analysability of darknets

2 Peer-to-peer Networks and Darknets

2.1 Peer-to-peer (P2P) Networks

In this section a short history of the internet leading to the development and spreading of peer-to-peer network architecture is given. The key concepts and properties of peer-to-peer systems are explained. It should provide a basic understanding of differences and advantages of the peer-to-peer network design.

2.1.1 From classical client-server architecture to distributed networks

The internet emerged from several military and sience research networks, with ARPANET as the most commonly known of them. It was planed and designed as a decentralized telecomunication network resiliant to outages. Although its resiliance on a network level, most its services have still a centralized structure. On failure not the whole network will fail, but a service, as a website like www.tu-darmstadt.de or its email system, can come unavailable easily.

This originates from the in most protocols and services used client-server architecture. The clients tries to find a single server to send its request to. The server processes the request and sends the response back. If in the time between the server is chosen by the client and it sends the response the server fails, the reqeust fails too. There are several methods in protocol and network design to prevent such failures.

The basic approach is to devide the responsipilities in different independant domains. For example an email server is not responsible for all the emails in the world but only for those of one (or more) domains. Note that such a domain does not necessarily mean a domain name in the DNS system but can be an arbitrary domain like a company, an university or just a group of people.

The next level of reduction of responsibility is to reduce what a single component is responsible for. Avoidance of single-point-of-failures, redundancy, load balancing where single requests are distributed to different servers are the common ones. However, all the approaches on this level scale very badly and therefor are highly expensive when achieving good fault tolerance or dealing with large amount of clients or reqeuests.

Taking the splitting of responsibility to the maximum, every one is responsible for everything. This is basicaly the idea behind the peer-to-peer architecture in which every client is simultaniously a server. Of course not every client is responsible for everything in a network but each of them is participating in the service of the network, serving a fraction of the serving while for each fraction there are multiple servers. Since there is commonly no distinction between clients and servers a participant is called a node or peer.

2.1.2 The rise of P2P systems and todays usage

With the Domain Name System (DNS) and the Simple Mail Transfer Protocol (SMTP) some parts of the peer-to-peer design are already used since early days of the internet. More than that, P2P ideas were researched, developed and used in smaller scale for a long time. But it took until the late ’90 and file sharing networks as Napster to popularize the P2P concept.

This high usage for unauthorized file sharing of copyright protected material led to a infamous image. Under this the term P2P suffered for quite a time. Just in the past few years several larger companys began to utilize peer-to-peer systems, e.g. for delivering larger sets of data to a huge amount of customers.

Today P2P systems are more and more used as a resiliant and scalable basis for communication, content delivery, and distributed storage, both legaly in public and unclear or even illegaly in twilight.

2.1.3 What a peer-to-peer overlay is

Of course not every node in the internet or another network participates in a P2P network. And not every two peers in a P2P network have to be directly connected and be able to communicate with each other. Therefor the P2P network graph is only a subset of the underlaying topology graph.

Commonly a seperate addressing scheme for communication in such a network is used. In most cases a node is identified by just a sequence of bits, its ID. It is usually represented as a number in decimal or hexadecimal form. All possible IDs together are called the address or ID space and its size vary from network to network but is constant within one network.

Both these propertys together form a more or less idependant network put on top of the underlaying one. It is then called a peer-to-peer overlay.

2.2 Darknets: privacy preserving P2P overlays

In the last section we gave a brief overview of the key aspects of peer-to-peer systems and their envolvement over time. As any tool and technology, P2P systems have advantages and disadvantages and can be used for good or evil. Now we will discuss how the demand for privacy led to a more specialized class of peer-to-peer networks, the darknets.

2.2.1 Consequences of decentralisation

As discussed before the essence of P2P systems are their decentralisation. Not a single server but virtually every pariticpant of the system is responsible for serving requests which results in a high failure resiliance. What is a valuable property for its users can be problematic if trying to stop the service of a network. Preventing the unauthorized distribution of copyright protected material just as repressed communication is rendered nearly impossible.

TODO:weiter ausführen? eventuell 2 absätze, sonst mit nächster subsection zusammen legen?

2.2.2 The demand for privacy preservation

The distribution of objectable information can therefor not effectivly be prevented but the nodes could still be prosecuted afterwards if their membership and identity is revealed. In the used file sharing networks the communication between two nodes is indeed encrypted and not readable buy outstanders, but in order to exchange infromation the nodes have to connect to each other. Therefor their identity, in the internet commonly the used IP address at a given time, is revealed to each other.

This is, however, not limited to file sharing networks. In every regime undesired communication could be somewhat pushishable, regardless of it is a whisle blower in a missbehaving company or regime critics under a dictatorship.

TODO:erklärung/begründung warum trotz dual use solche technologie entwickelt und bereit gestellt werden sollte

For those oppressed environments a more privacy preserving class of peer-to-peer networks has evolved, the darknets. In the following the main differences to classical P2P systems are touched while they will be explained in detail in chapter 3.

2.2.3 Trust relation base membership and membership consealment

Exchanging any kind of information on the internet leads to the necessaty of directly connecting of at least two participants. As explained, this leads to revealing the identity of those in the underlaying network to each other. To overcome the impact of this privacy relevant information disclosure, connections are only established between mutual known and trusted parties.

Additionally to communicating with kown and trusted peers only, no information about which peers a node is connected to are passed. On compromisation only the identiy of directly connected peers are affected. Anyway, no node can be held responsible for contact to other nodes, since not even participants know to which peers a node is connected to.

2.2.4 Forwarding requests and hop-by-hop anonymity

But since only communicating with trusted peers would end in a very limited reachability, communication between not directly connected nodes has to be forwarded on some way. Thereby the identity in the underlaying network and information about the topology, who is connected to whom, have to be concealed.

To achieve this, forwarded messages are modified to originate from the forwarding node itself. Returning answers are modified accordingly and are passed back to the source of the original request. As this is done on every node, the identity of the origin of a message is preserved and no information about the topology is reveald.

2.3 Darknet characteristics and resulting challenges

The methodical differences of darknets to classical P2P networks were explained. Now follows, althoug already roughly touched, a discussion of the arising characteristics and the thereby resulting challenges for the practical use of darknets.

2.3.1 Hidden Topology

TODO:mehr...

In summary the membership of a node in a darknet is only known to its trusted peers and messages can originate from either the node they are received from or any node beyond it.

2.3.2 Difficulties for routing ...

This high rate of protection of privacy relevant information comes with numerous difficulties in designing and evaluating simultaneously resilient and salable darknets. Messages whose destination is not within a nodes neighbors have to be forwarded to some nodes in between. In conventional networks the next node can be chosen on the basis of topology information about the network, e.g. in form of a classical routing table or structured overlays in P2P systems.

Though any topology information about the network is confidential, they are not distributed and collected by the nodes and not available for deciding to which node a message is given next. This holds as well for meta topology information such as the origin of a message and therefore the direction the according node lies within.

2.3.3 ... and evaluation

TODO:mehr... The same difficulties arise for measuring any quality in a darknet, for example for evaluation and comparison of decisions while development.

2.4 Metrices for routing evaluation

2.5 Evaluation methods for networks

For evaluating distributed systems like P2P overlays, and darknets in particular, there are four different approaches.

TODO:...mehr

2.5.1 Full blown test environment or mathematical model evalution?

The original software can be tested, almost or completely unmodified, in a testbed. This is used to test the functionality itself but scales very badly for multiple nodes and even more for larger networks. In particular for evaluating darknets, information gathering methods have to be added since this is against their normal usecase. TODO:vor und nachteile ausarbeiten

A completely different approach is to build a analyical model and derive formulas for the relevant values from it. This is quite flexible for varying parameters and very scalable and fast. But on the other hand the derivation of formulas can be challenging and small changes in the model or algorithm can render most of the work inappropriate. With analytical models upper and/or lower bounds can be estimated, but real world performance can depend on protocol details and other factors hard do model.

2.5.2 Inbetween: emulation and simulation

By removing all irrelevant and independent parts of the original software and run many instances of such slimed clients a network can be emulated. Main benefit is that no new software has to be written that can be faulty or not sound to the original one. While very little new implementation effort is required, in common the client has still much overhead e.g. for network communication.

This consummes much more resources in time and memory than a simulation, whereby the abstract, relevant behavior is implemented in a simulation environment. Large networks and complex algorithms can be simulated. Implementation is strait forward and can be easier then building a proper formula. However, the algorithms must soundly be implemented and simulation can take much computation time and memory depending on the model and the network to be simulated.

2.5.3 Applicability on darknets

2.6 Survey of previous darknets

3 Model Description

3.1 Why we chose the simulation based approach

As explained in Chapter 2.4, different methods to evaluate darknet systems exist, each suitable for different usecases. To analyse a routing algorythm, its performance, strengths, and weaknesses, larger scale networks are neccessary. For this, especially while dealing with changeing algorythms and parameters, the simulation approch is most appropriate since it has the best tradeoff of flexibility and scalabiliy. However, this method is up to now underrepresented. To change this we decided to develop an easy to use and extend model and implement it in an wideley adopted network simulation framework as OMNet++. TODO:how to prove this statement? This gives the posibility to easily estimate the behavior of a routing algorythm in a darknet while maintain comparability of the results.

3.2 Node based with fixed neighbor set and add-hoc anonymity (static)

Virtually every P2P client supports a basic set of capabilities. These are connecting to other clients, storing and recieving of information and some kind of searching. Depending on the network searching for files or other nodes can be different or the same. Naturally these reqests can or even have to be responded to.

So on an abstract level, a P2P node has to be able to connect to other nodes, make requests and send an response to a received request. The basic model used in this work does not distinguesh between the different request types since in most cases it makes no difference for routing algorithms. This is the same for darknet nodes. Only the peers a darknet node connects to or accepts connections and messages from of course are limited.

In practice transfering the real world trust relation to the darknet and the distribution of the current addresses of onces peers are two of the hardest tasks. There are some fundamentaly different posibilities to solve this depending on the overall design. But since these tasks are better analyzable than routing algorythms our model excludes these topics and concentrates on routing makeing it easyer to evaluate this seperate problem.

Additionally a darknet client has to be abled to forward reqeuests to other nodes. In darknets the hop-anonymity is typically achieved by modifying forwarded messages as to come from one self. Responses to forwarded requests have to be modified and forwarded respectively to the origin the reqeust was received from.

This simple model is static, all nodes are always online and a failing node is not possible. It is very simple and scalable because only a minimal overhead is needed. But it is not very realisitc since nodes are not online all the time and can fail on one way or another.

3.3 Churn model extension with bootstrapping and offline detection (dynamic)

To take into account the posibilitie of nodes shutting down, having connection problems or failing otherwise, the model is extended by a churn based lifetime model. It gives the nodes a probabilistic lifetime, so not all nodes are online in the begining, but they bootstrap into the network. They have to establish a connection to their peers.

Nodes online/offline state can change during the simulation, so the peers of this node have to be informed. As this simulator concentrates only on the routing algorythm we are not concerned how a node would detect a change of state of one of its peers. The simulator notifies all peers of a node of the change. In the real world something like an TCP-like acknowledgment method would be used. TODO:change code: model it by offline/online notification from simulation environment

3.4 (??)Extension of the model by example

4 Implementation

4.1 The simulation library and framework OMNeT++

4.1.1 Alternatives and why we cose OMNeT++

4.1.2 What is OMNeT++

OMNeT++ is an simulation library and framework written in C++. It is modular and extensible and primarily used to simulate networks not only limited to telecomunication networks. The simulation

4.1.3 The simulation process

4.1.4 Measurement collection

4.1.5 INET: The communication networks simulation package

4.2 Implementaion of the static model

4.3 Churn based lifetime model

If a node goes offline it has to be detected by its peers. Therefor, an acknowledge mechanism is needed. Basically a acknowledge message (short: ACK) is send back on receieving a message. If such an ACK is not received within a certain amount of time, the message can be resent up to a selectable number of times. If no ACK is received after that, the peer is no longer considered the be online and connected.

4.4 two simple example routing models

4.4.1 randomwalk

- with n-degree fanout
- very simple loop prevention

4.4.2 flooding

4.5 notes

kapite aus theoretischem model auf implementierung (bei kap.4.2) matchen?
probleme vor denen wir standen/ zu erfuellende requirements: beschreiben wie metriken ausgewertet werden

5 Simulation and Evaluation

5.1 Simulation environment

5.2 Used metrices (nicht wirklich hier; eher in implementation)

- (average/max) path length
- sent message count
- faild routings / requests OR droped packages

5.3 Q: Scalability of the model/framework

Used RAM; RAM RAM and moar RAM (sprich: metriken die nicht in darknetzen anfallen also nur fuer simulation relevant sind kurz erklaeren)

5.4 Q: Impact of fanout degre at randomwalk on found pathlength

comparison to flooding which finds shortest path

5.5 Q: Probability of path failure on return for churn model

6 Conclution and Future Work

- Everything is epic, but.. ;)
- Extend OMNet model to take the underling network into account
  - real path length; relevant if protocol tries to take it into account but difficult in a darknet environment
- Implement tested darknets and compare to their tests
- Implement untested/new nets and improve routing parameters
  - or even decide if algorythm is pratical usefull or not