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 Darknets

In this chapter the concept, the characteristics and the thereby arising problems of darknets are explained. Metrics for darknet evaluation are discussed and existing darknet approaches are surveyed.

2.1 What is a P2P network, what a darknet and how do they diver?

In modern computer networks not only the classic client/server architecture is used but also peer-to-peer (P2P) networks have gained importance. In this concept every client takes additionally a server-like role and distributes information to others clients, also called peers. Although probably the main driving force for its development was the use of file sharing, today P2P is adapted for communication, content delivery and multimedia streaming.

But in conventional P2P networks neither the membership in the network nor the stored or requested content is concealed. Since the demand for privacy and anonymity preserving communication channels has risen, some P2P concepts with respect to these needs have emerged. In such requirement needing environments like regime critical or whistle blowing communities it is essential that the membership in such a network and even more importantly the activities in it are kept secret.

This includes the untracability of both the requester and the publisher of files and the location of files in the network. Even the meta data of files like the file size have to be protected since they may allow conclusions which content is requested. P2P networks respecting these requirements are called darknets.

Therefor darknets are P2P overlays in which participants connects and communicates only to others they have some trust relation with. Nodes do not pass to whom they are connected to. This leads to only a minimum number of trusted participants knowing of ones membership in a darknet.

Requests for storing, searching and receiving files are forwarded to other nodes but are modified to look as they come form one self. Same is done accordingly with responses, or, more general, to all messages. Thereby no node on such a chain of forwarding nodes can tell if a message originates from the node it received it from or any node beyond it. Any data distributed in the network is chunked in same-sized parts which are addressed and distributed individually. Furthermore as they are encrypted no node storing or forwarding them knows what files they contain.

2.2 Implications of darknet characteristics

In summary the membership of a node in a darknet is only known to its trusted peers, the files stored on a node are unknown to and unreadable by this node, and messages can originate from either the node they are received from or any node beyond it.

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 metrices for routing (specially in darknets) evaluation

The same difficulties arise for measuring any quality in a darknet, for example for evaluation and comparison of decisions while development. In general several metrics exist for measurement and comparison of routing algorithms. But since we utilize an abstract model of darknets we consider basic metrics as they apply to all kind of networks.

- The simplest metric is the path length, or hop count, the amount of hops a packet has to be forwarded on until it reaches its destination. It is an important factor of delays in communication and also affects the bandwidth between nodes. The shorter the chosen path is, the faster the communication is and the less the network has to be utilized. The path length, and its average and maximum in a network, are the most commonly used metrics to compare routing.

- The overhead measures how much resources have to be used to transmit the actual information. It can be measured as the overhead messages ratio or the overhead bandwidth ratio and is a grade for the efficiency of a system.

- In one simulation, we will inspect the count of failed return paths. Since a response is sent back the path it came from, it has a fatal impact if a node on that path fails. It can show a upper bound of reliability of a system if it relies on this method.

- TODO:more?

2.4 Available measurement methods for darknets

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

The most theoretical one 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 hard do model.

A more realistic approach is to simulate a client based on the model. The abstract relevant behavior is implemented in the simulation environment. Large networks and complex algorithms can be simulated while implementation is strait forward and 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.

Commonly less implementation effort is needed when emulating a network. This is commonly done by removing all irrelevant and independent parts of the original software and run many instances of such slimed client. Main benefit is that no new software has to be written that can be faulty or not sound to the original, but in common the client has still much overhead e.g. for network communication. This consummates much more resources in time and memory than a simulation.

The original software can as well be tested, almost or completely unmodified, in a testbed. This is used to test the functionality itself but scales very badly for multiple nodes or even larger networks. Particularly for evaluating darknets information gathering methods have to be added since this is against its normal use.

2.5 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 What is OMNeT++ and why we chose it

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

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