3.1 Unstructured Peer-to-Peer Networks

3.1.1 Which mechanisms/approaches could be used?

The following different approaches can be considered:

1. Reflector Node:
   Each Query Hit message is cached, so if the same subsequent Query message should be answered the reflector can use the answer from before reducing overflooding.
2. Randomized Forwarding:
   Each Query message is not sent to all immediate neighbours but a random set so we can reduce overflooding.
3. Distributed Flow Control:
   Nodes store and exchange information about the current load and can redirect requests to lessen the load on nodes.
4. Establishment of efficient overlay network topologies by considering underlying topology:
   Assuming that all nodes are within a domain (there can be several domains). If we know the topology and want to connect to another domain then by connecting to a node which has a connection to the other domain, the other domain can be reached with the lowest amount of messages which also reduces overflooding within our domain.
5. Hierarchical network organization (e.g. Kazaa):
   Every peer is an ordinary or super peer, and every ordinary is assigned to a super peer. The super peer are organizing themselves and protect the ordinary peers from traffic which reduces the chances of overflooding. The super peers forward queries from their own ordinary peers to other super peers and answer queries from their own ordinary peers and other super peers.
6. BitTorrent:
   A file is distributed over many different peers (swarm). A tracker or coordinator knows which peers have which file so if any client requests a certain file the tracker will sent a list of peers which contain the file. This implementation avoids bottlenecks.

3.1.2 Could SDN be used to optimize?

Yes, SDN can be used to optimize a P2P-network because its controller has an overview over the load on each node and can redirect transmissions such that the chance of overflooding can be reduced.

3.2 Advantages and Disadvantages of Mapping in P2P networks over conventional DNS mapping

Because the conventional DNS mapping is centralized it is highly dependent on the centralized server and if this specific server fails the whole mapping system will break down. In P2P networks the mapping is distributed and therefore not so prone to such failures. Furthermore P2P networks have better reliability with higher latency and robustness.
The disadvantage of mapping in P2P networks is that more network ressources are required and that in pure P2P networks the scalability is lessened.

3.3 Advantages of the tracker in BitTorrent P2P networks over Distributed Hash Tables

Because Distributed Hash Tables is decentralized there is no central authority as the tracker is in the BitTorrent implementation. Furthermore BitTorrent is much more efficient and therefore quicker (lower latency) because the Tracker gives a list of peers which have the file whereas DHT first needs to search for the file. Also DHT has a huge network traffic overhead it is less able to scale up. Last when a node fails or new nodes join the BitTorrent implementation is much more robust in handling those cases than Distributed Hash Tables.

3.4 Structured Peer-to-Peer Networks

3.4.1 Finger Table for Nodes 0 & 19

Node 0	
N0 + 1	N1
N0 + 2	N5
N0 + 4	N5
N0 + 8	N11
N0 + 16	N17
N0 + 32	N37

Node 19	
N19 + 1	N23
N19 + 2	N23
N19 + 4	N23
N19 + 8	N29
N19 + 16	N37
N19 + 32	N53

3.4.2 Route from node 1 to 54

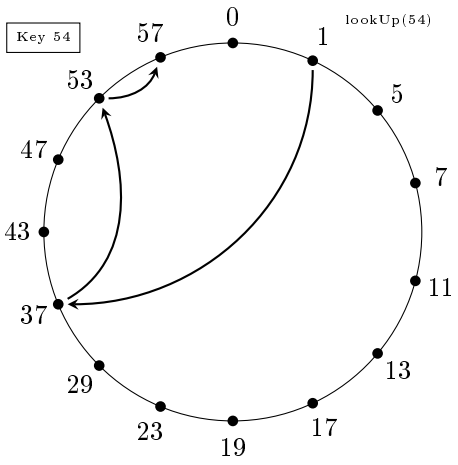
Finger Tables of the required Nodes

Node 1	
N1 + 1	N5
N1 + 2	N5
N1 + 4	N5
N1 + 8	N11
N1 + 16	N17
N1 + 32	N37

Node 37	
N37 + 1	N43
N37 + 2	N43
N37 + 4	N43
N37 + 8	N47
N37 + 16	N53

Node 53	
N53 + 1	N57
⋮	⋮

Diagram with Routes



3.5 How does a specific P2P network deal with high churn rates? (Example Chord)

A high churn rate means that many nodes join or leave the network which can lead to huge problems within a P2P-network. In a Chord implementation there exists the following algorithm for nodes joining the network:

1. Initialize node **n** (the rpedecessor and the finger table)
2. Notify other nodes to update their predecessor and finger tables
3. The new node takes over its responsible keys from its successor

If many nodes leave the network it is still not a problem because every node records a whole segment of the circle adjacent to it, especially it knows the nodes preceding and following it. Therefore we can assure that the network functions even if some nodes disappear.