

Observations on KAD

Ernst Biersack
Institut Eurécom
Sophia-Antipolis, France
March 2007

Joint work with Moritz Steiner and Taoufik En-Najjary, Institut Eurécom

Outline

- What is a DHT/KAD
- How KAD works
 - ◆ Routing
 - ◆ Publishing
 - ◆ Searching
- Crawling KAD
 - ◆ Results for full crawl
 - ◆ Results for partial crawl
- Spying on KAD control traffic
 - ◆ Keyword and file popularity
- Attacks on KAD
- Conclusions

What is a DHT

What is a DHT

- ◆ A distributed database for publishing and searching information
- ◆ Consists of many peers, each one is responsible for storing part of the database
- ◆ How to partition content of DB
 - ☞ Use key of the object to decide on which peer to store information
- ◆ What is a key: unique identifier
 - ☞ Key = hash(IP@), or
 - ☞ Key = hash(string)
- ◆ Each peer and each object is identified by its key

Example: Each key k is a bit string of length $m = 128$ bits:

XOR metric:

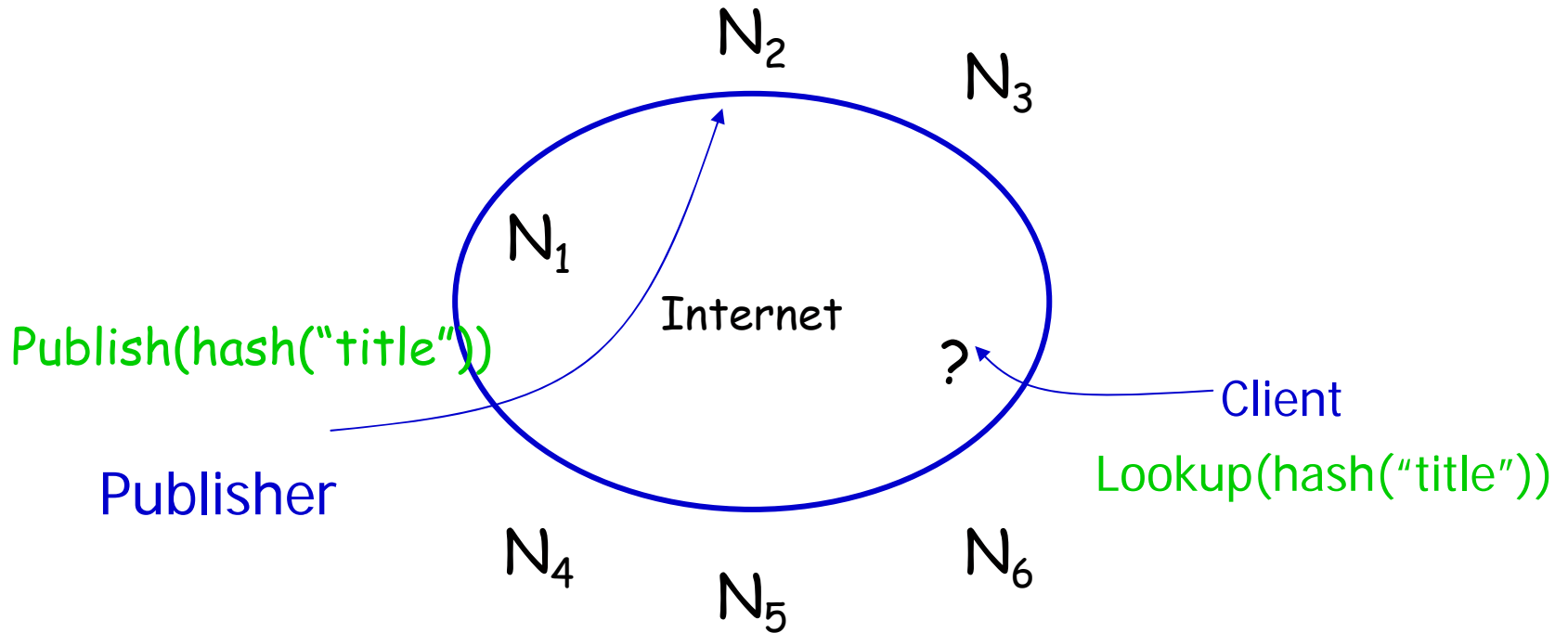
◆ Let node $j = j_{m-1}j_{30} \dots j_0$ and key $k = k_{m-1}k_{30} \dots k_0$:

◆ Note that closest ID is **unique**:

$$\text{☞ } d(j,k) = d(j',k) \Leftrightarrow j = j'$$

$$d(j,k) = \sum_{i=0}^{31} |j_i - k_i| \cdot 2^i$$

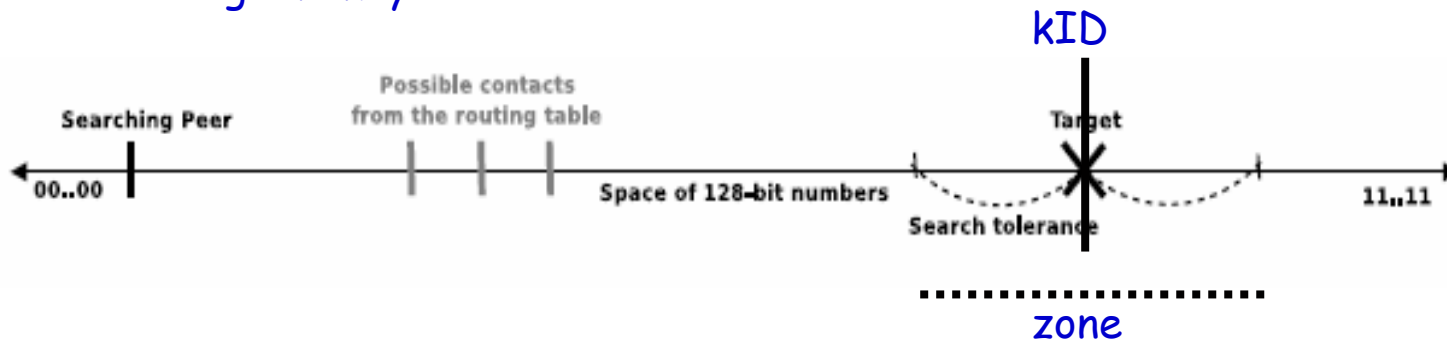
What is a DHT



- Important issues
 - ◆ How to partition key space
 - ◆ How to route
 - ◆ **How to survive churn (peers joining and leaving all the time)**

KAD: How to Publish

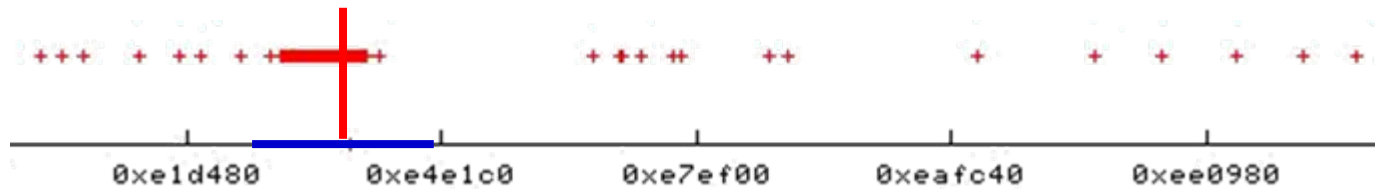
- Where to store the information for a given a key **kID**?
 - In Chord: On the first node who's ID is larger than **kID**
 - In KAD: On a node, who's first 8-bits are the same as **kID**
 - i.e. there is a zone and all peers in this zone are potential candidates for storing the key



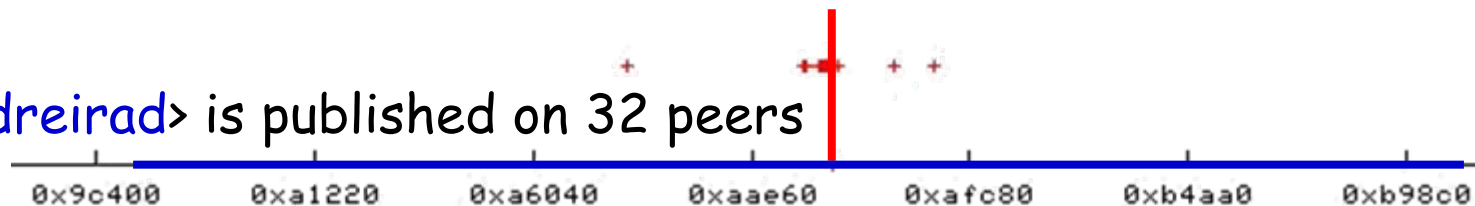
- A zone defined by the first 8 bits is $1/256$ of the entire key space, and contains a **few thousand peers**
 - How to find later a key (contact several thousand peers?)
 - Each key the information that comes with the key is stored $r=11$ times (**on 11 different peers in zone**)
 - This high replication assures that
 - Key is not lost when one peer leaves

Keyword Hash Distribution from Measurements

- Distribution over the KAD hash space for a popular keyword and an unpopular keyword
- The popular keyword is spread on 30 times wider keyspace
- `<the>` is published on 594 peers (99.7% inside the 8bit zone)



- `<dreirad>` is published on 32 peers



Exploring KAD

- How
 - ◆ Crawl
 - ◆ Spy

Full Crawl

■ How

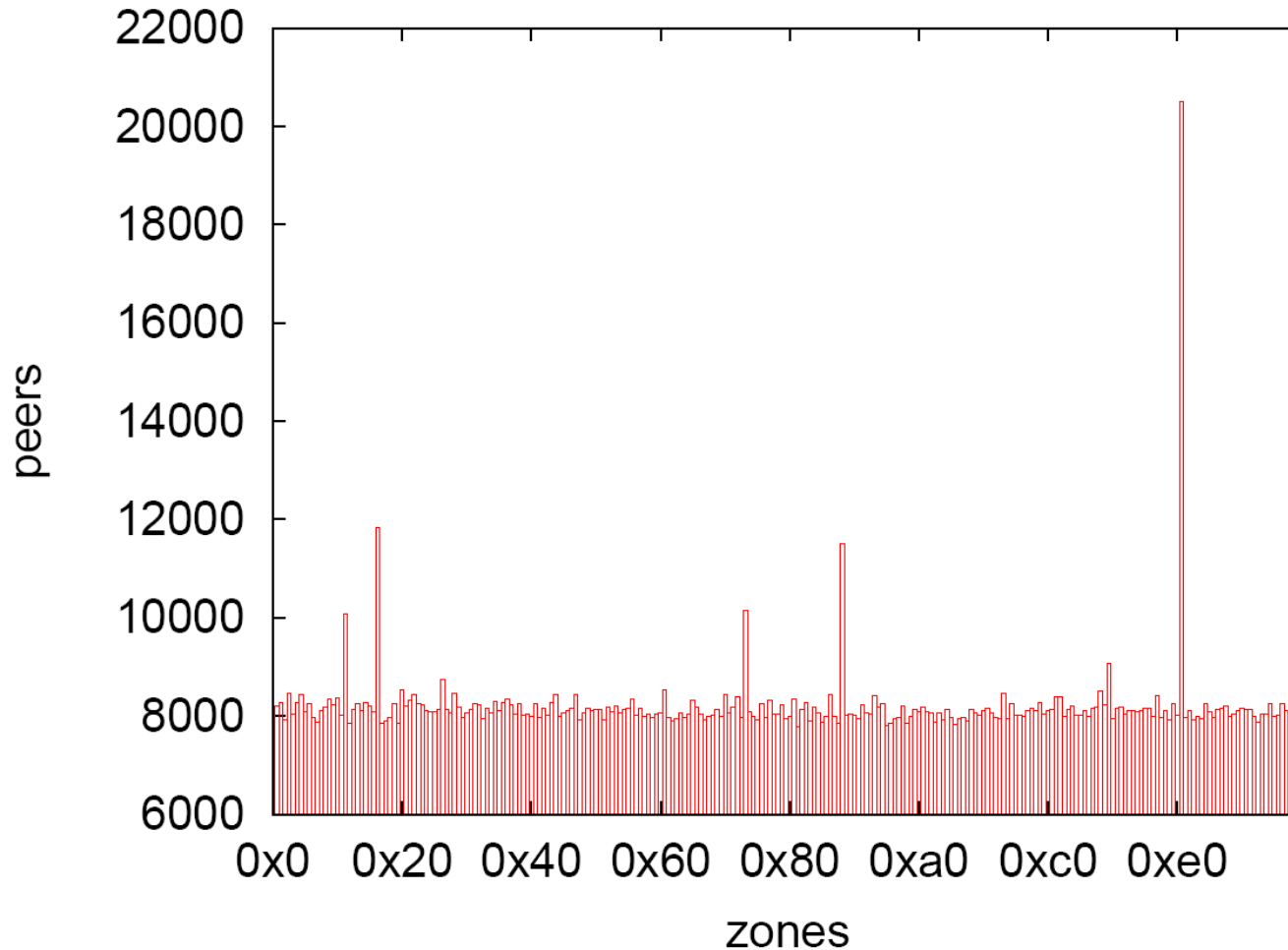
- ◆ Use only KAD route request packets
 - ☞ Considering the routing tree structure query for hashes
 - ☞ Answers contain peers close to the hash asked for
- ◆ Query the peers obtained in answer from the last round
- ◆ Do not saturate peers by sending too much queries
- ◆ Query each peer only once

■ Full crawl

- ◆ Takes about 20-25 minutes
- ◆ 17-22 GB traffic
- ◆ Saturates a 100 Mbit/sec link at Uni Mannheim

- ◆ 3 to 4.3 million peers *seen*
- ◆ 1.5 to 2 million peers responding

Full Crawl: Distribution over the hash space



Except for the modified clients, the peers are uniformly distributed over the hash space.

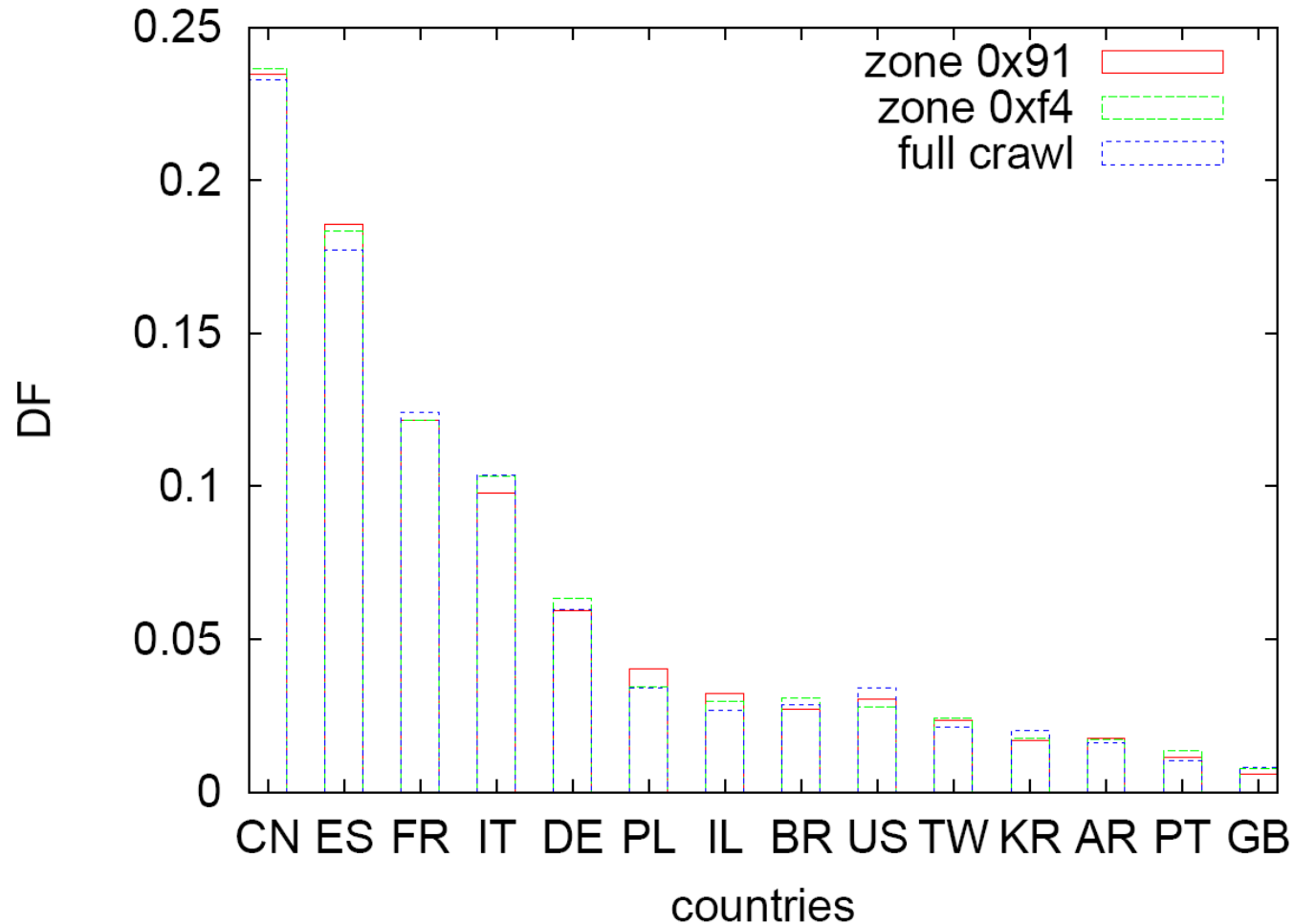
Full Crawl: Anomalies

- One (or few) IP @s and several (up to 100'000) *hand* chosen KAD IDs
 - professional crawler (in LA area)
- Many different IP @s (from one language area) all with the same KAD ID
 - modified client

Partial Crawl

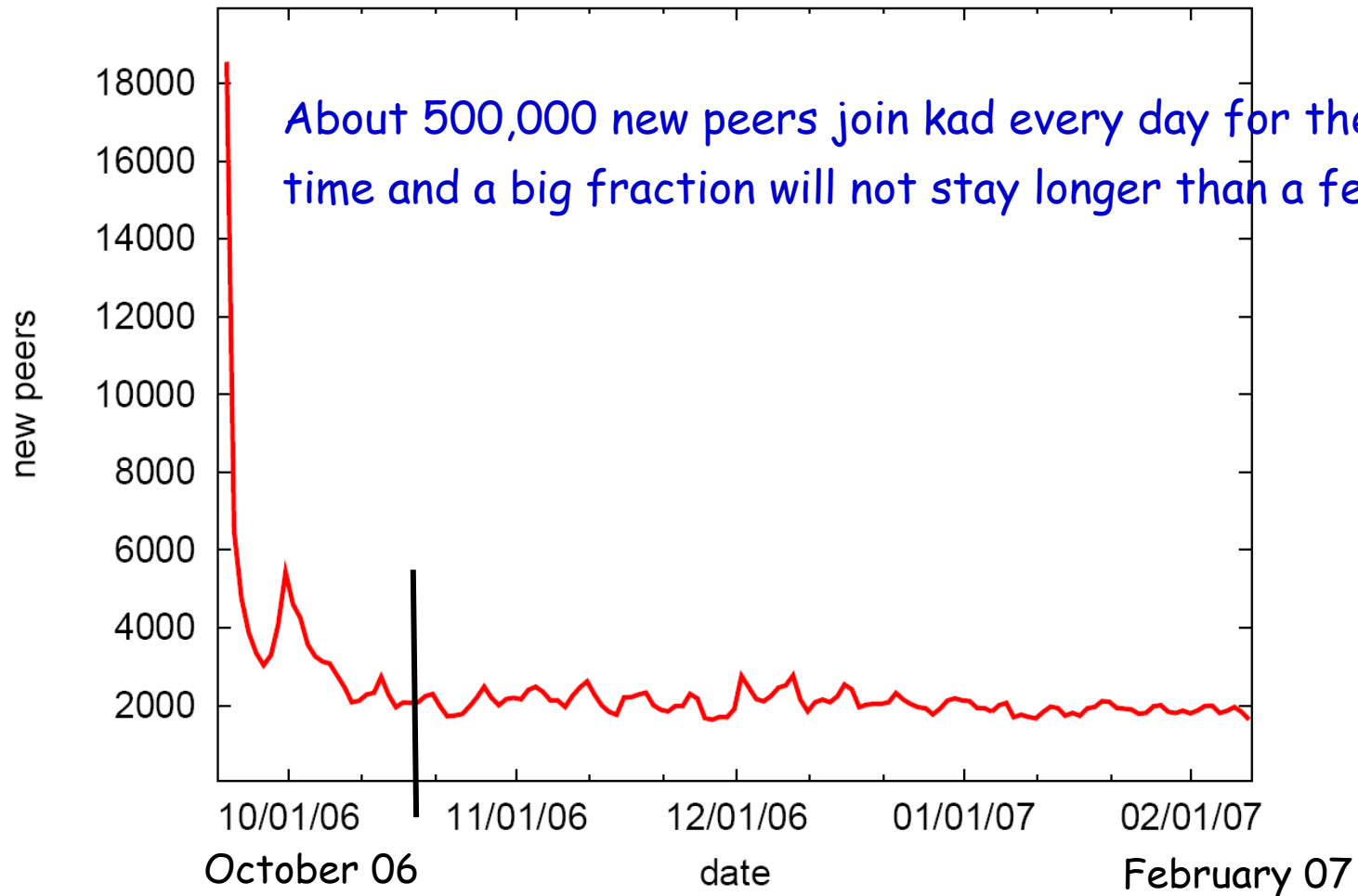
- Why?
 - ◆ Less expensive than full crawl
 - ☞ Can do it more often
 - ◆ Due to hashing, partial crawl should give representative picture of the system
- How
 - ◆ Crawl 8-bit prefix zone ($1/256$ of the entire hash space) containing 15,000 - 25,000 peers
- How fast
 - ◆ 2.5 seconds crawl time (+some seconds waiting for late responses)
- How often
 - Every 5 minutes
- For how long
 - ◆ Since September 2006, more than 180 days
 - ◆ Without crashes!
 - ☞ Two independent crawlers at Uni Mannheim and Eurecom
 - ☞ Observations are "merged"

Crawl: Geographic Distribution



The geo. distribution of the peers of the full crawl is close to those of an 8-bit zone crawl.

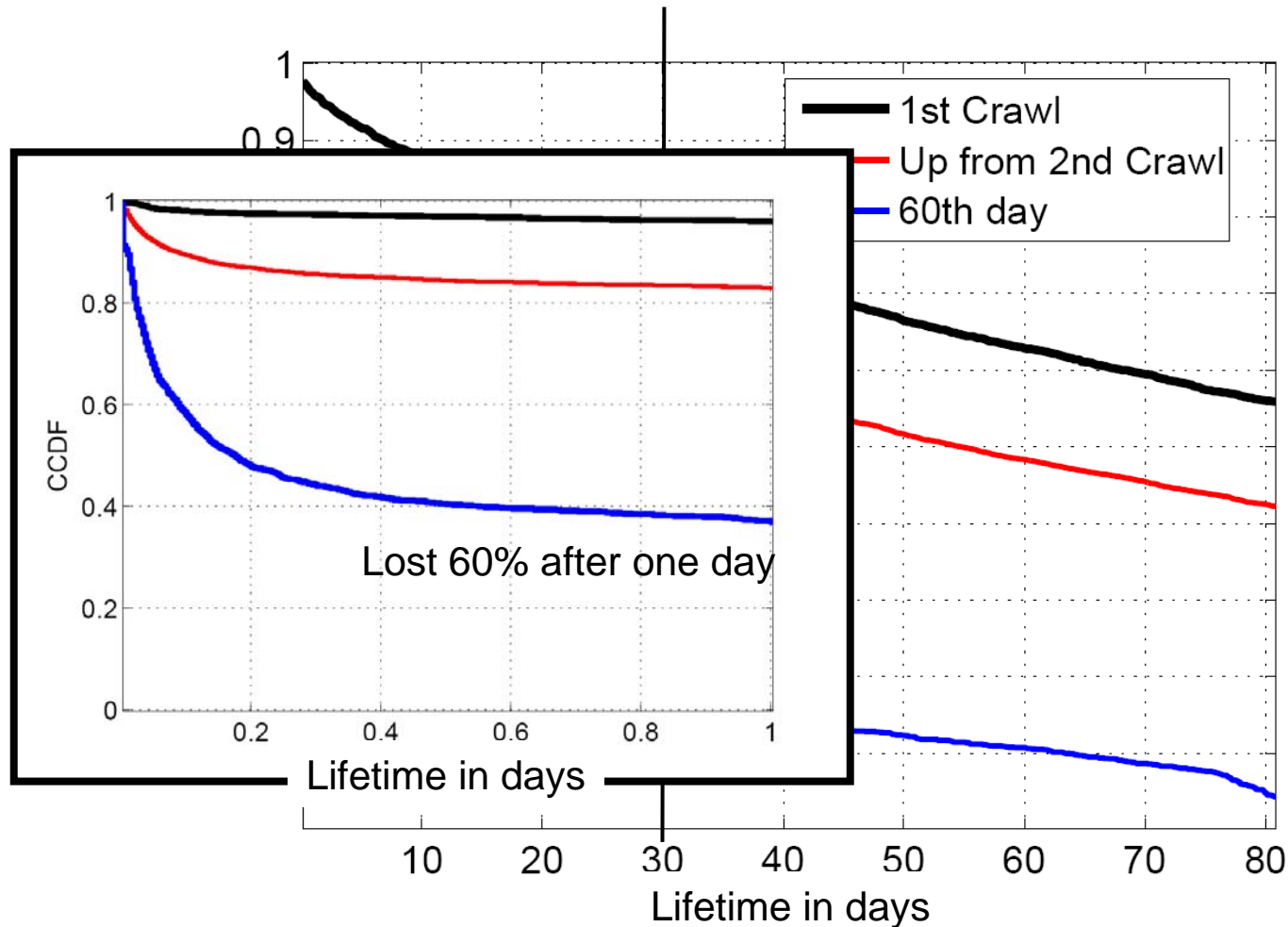
Partial Crawl: New Peers



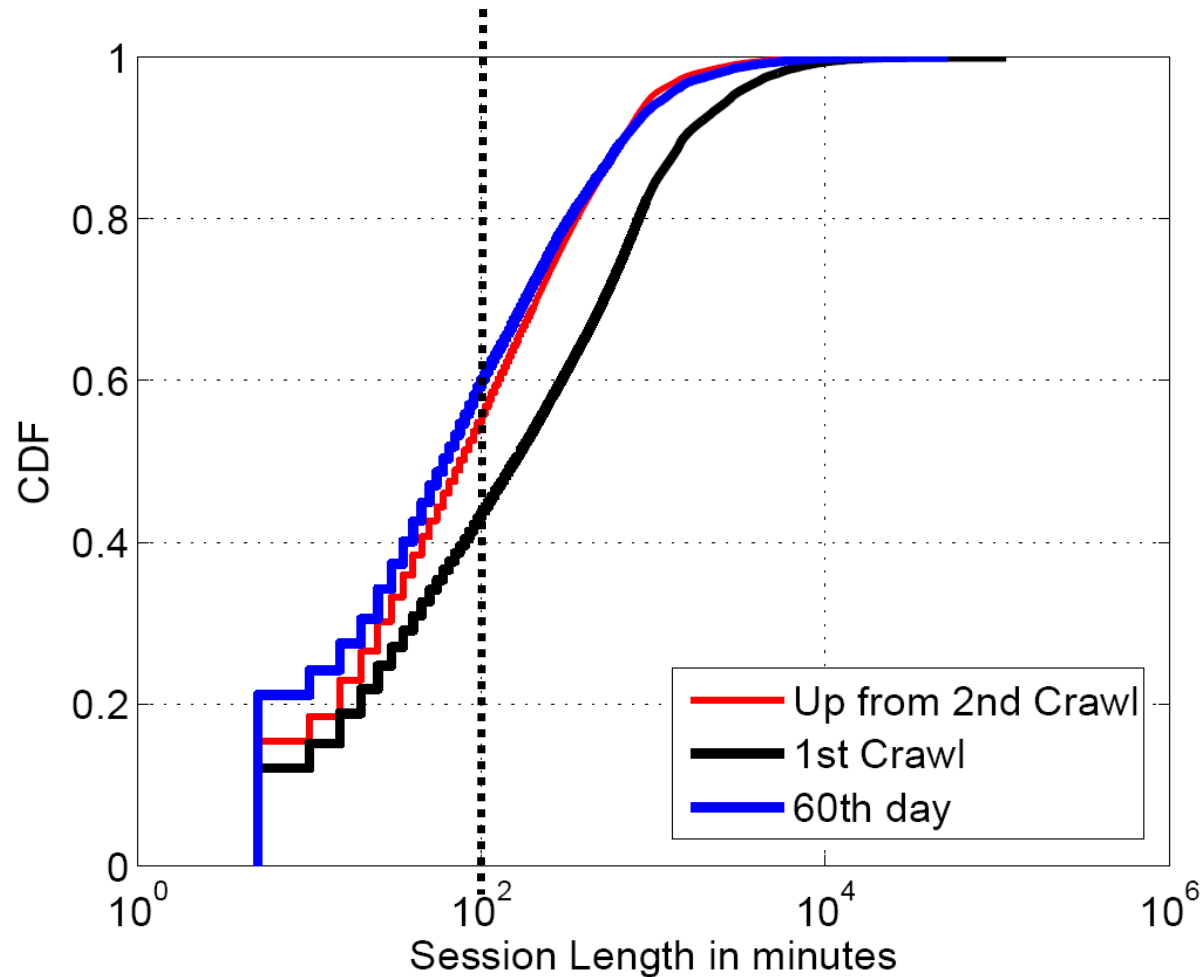
The number of peers never seen before stabilizes after some weeks.

Partial Crawl: Lifetime

Lifetime is the time between the first and the last appearance of a peer.



Partial Crawl: Session Length



At the first crawl we see a big fraction of peers that are part of to *stable core*.

Crawl: Conclusion & Outlook

- Developed the (today's) fastest crawler for the KAD network
- Data analysis still under way

- Observations
 - ◆ A small *core* of stable peers
 - ◆ A big fraction of volatile peers seen only for very short time

- Open issues:
 - ◆ Predict the behavior/stability of a peer

Spying on KAD

- Why
 - ◆ Find out what keywords are popular
 - ◆ How much traffic is generated
- How to spy on part of the hash space called **I** (e.g. 8-bit zone of all KAD ids starting with **xe3**)
 - ◆ Introduce a large number of spy peers that have KAD ids in **I**
 - ☞ Crawl **I** to find all the peers **N** with KAD ids in **I**
 - ☞ Announce the spy peers to the real peers **N** (to “poison” their routing tables)
 - ◆ How many spy peers?
 - ☞ As many to ensure that of every search and publish operation is seen
 - at least once, and
 - if possible not more than once (to avoid being too intrusive)
- Scalability of spy: All the spy peers are running on a single PC (a single program emulates as many spy peers as we want)
 - ◆ To reduce the memory requirements, no state (e.g. for publish) is kept
 - ☞ → Search requests cannot be properly answered, instead a fake answer is given

Spying: Control Traffic

- Spied on the 8bit zone <e3> during 12 hours

- Search

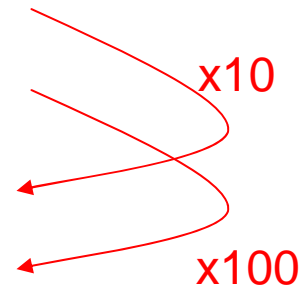
◆ Messages	561 542
◆ Traffic	10,8 Mbytes

- Publish

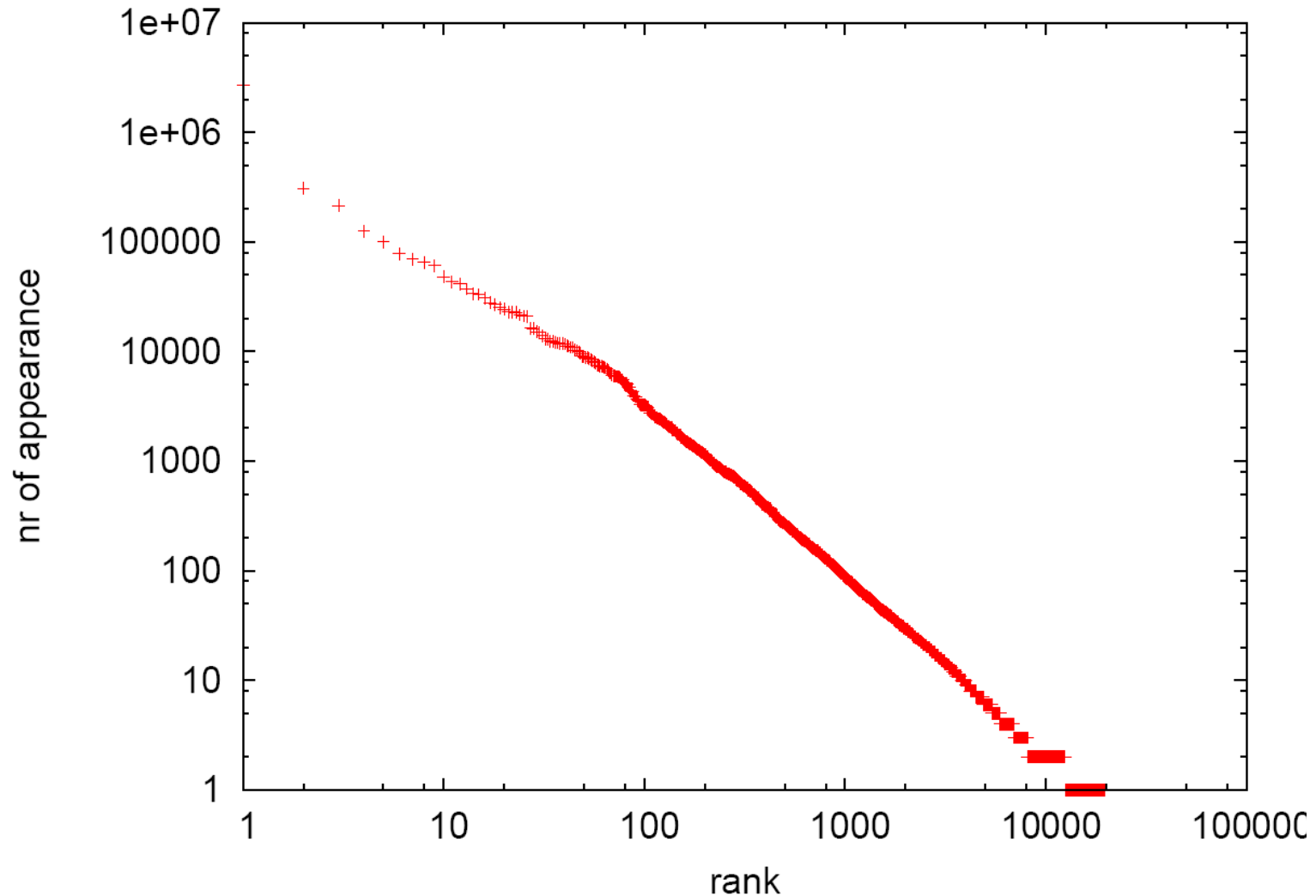
◆ Messages	5 549 183
◆ Traffic	966 MBytes

- Route

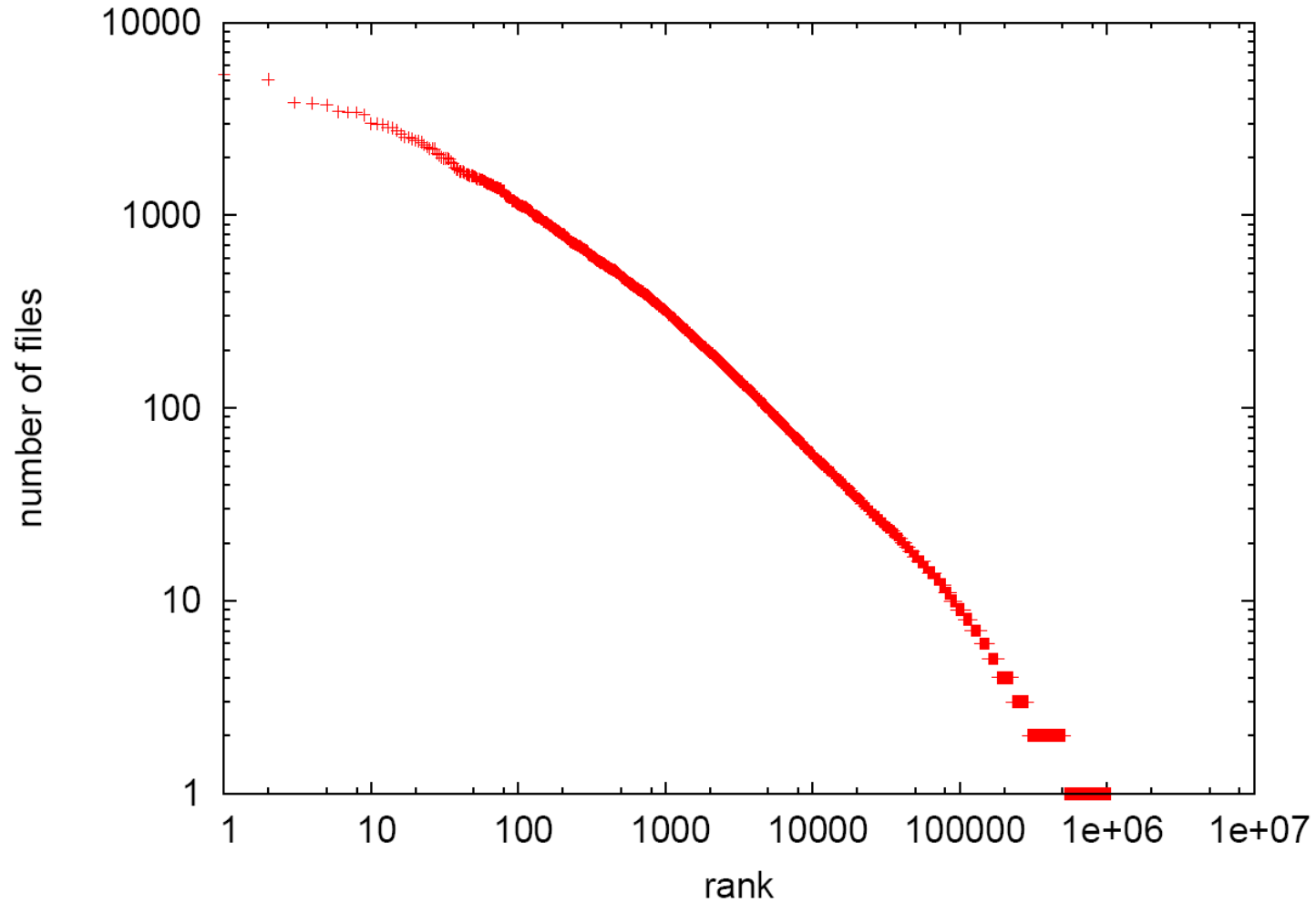
◆ Messages	9 761 278
◆ Traffic	342 MBytes



Spying: Keyword Popularity (cont')



Spying: File Popularity





Attacks in P2P

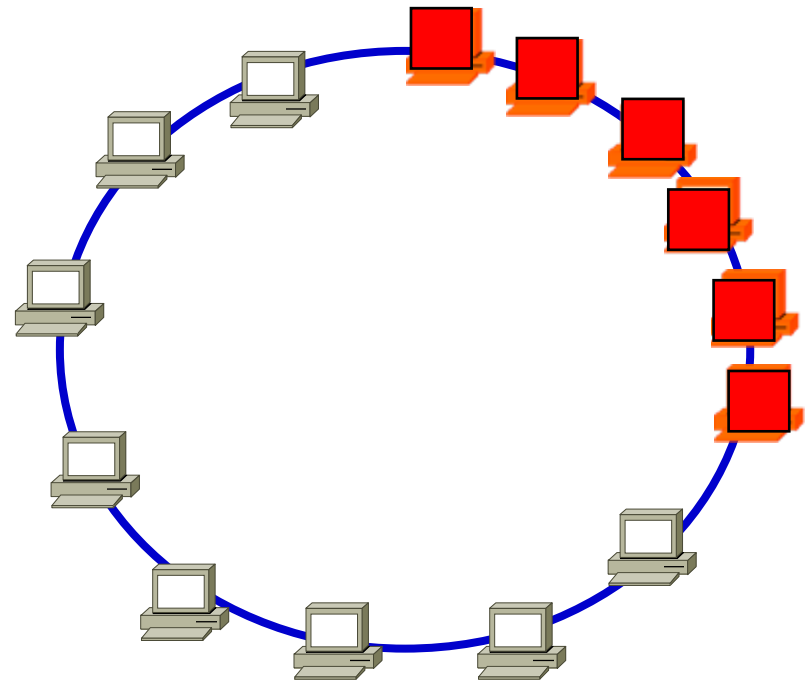
- Many attacks possible
 - ◆ Poisoning Attacks (files with contents different from description)
 - ◆ DDoS Attack

 - ◆ Sybil Attack
 - ◆ Eclipse Attack

Sybil Attack: Definition

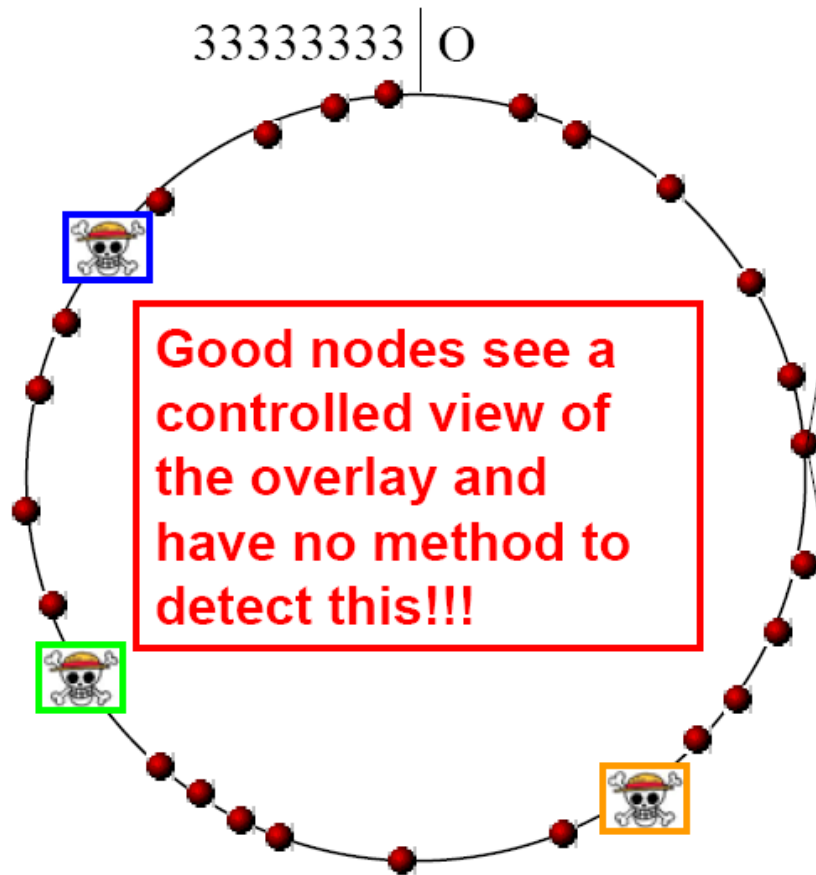
- Register many times with multiple identities
- A *Sybil attack* is the forging of multiple identities for malicious intent -- having a set of faulty entities represented through a larger set of identities.
- The purpose of such an attack is to compromise a disproportionate share of a system.

Douceur, J. (2002). The Sybil Attack. 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), February 2002. Online source URL: <http://www.cs.rice.edu/Conferences/IPTPS02/101.pdf>



Eclipse Attack

- ◆ Eclipse Attack
 - ◆ Malicious peers conspire to hijack and dominate the routing table of correct nodes
 - ◆ "Eclipse" correct nodes from each other
 - ◆ Control data traffic through routing



NodId 10233102			
Leaf set	SMALLER	LARGER	
10233033	1	1	10233122
10233000		10233230	10233232
Routing table			
0	1	2	3
10-0-31203	1	2	1
102-0-0200	102-1-1302	102-2-2302	3
10233-0-01	1	10233-2-121	3
0		102331-2-0	
		2	
Neighborhood set			
13021022	1	11301233	3
22301203	2	3	33213321

General Conclusion

- KAD is a great subject for study
- Many issues raised
 - ◆ High redundancy at all level results in large amount of traffic
 - ◆ Very easy to “eclipse” part or all of KAD
 - ☞ Is a decentralized solution really better than the old centralized one
 - ◆ Peers can be turned into DDoS bots
 - ☞ Tried it on ourselves
 - Over 100 Mbit/sec incoming traffic
 - ◆ Theory vs. Practice: Many theoretical DHT designs
 - ☞ Very few papers that address security
 - ☞ Not really any practical solutions against various attacks such as Eclipse attack