EE817/IS893 Blockchain and Cryptocurrency Peer-to-Peer Systems

Yongdae Kim

KAIST

Admin

- Student Information Survey
  - https://goo.gl/forms/VnjAyN5N1bmswLNP2
- Paper Presentation Survey
  - https://goo.gl/forms/pGhbDPJqBr4MNff92
- Paper Presentation vs. Reading Report Scoring
  - If you present a paper, you will be exempted from four reading reports.
- Project

P2P System: Definition

- A distributed application architecture that partitions tasks or workloads between peers
- Peers are equally privileged, equipotent participants in the application
  - Forming a peer-to-peer network of nodes.
- Peers make a part of their resources directly available to other peers
  - processing power, disk storage or network bandwidth
  - without the need for central coordination by servers
- Peers are both suppliers and consumers of resources

P2P Applications

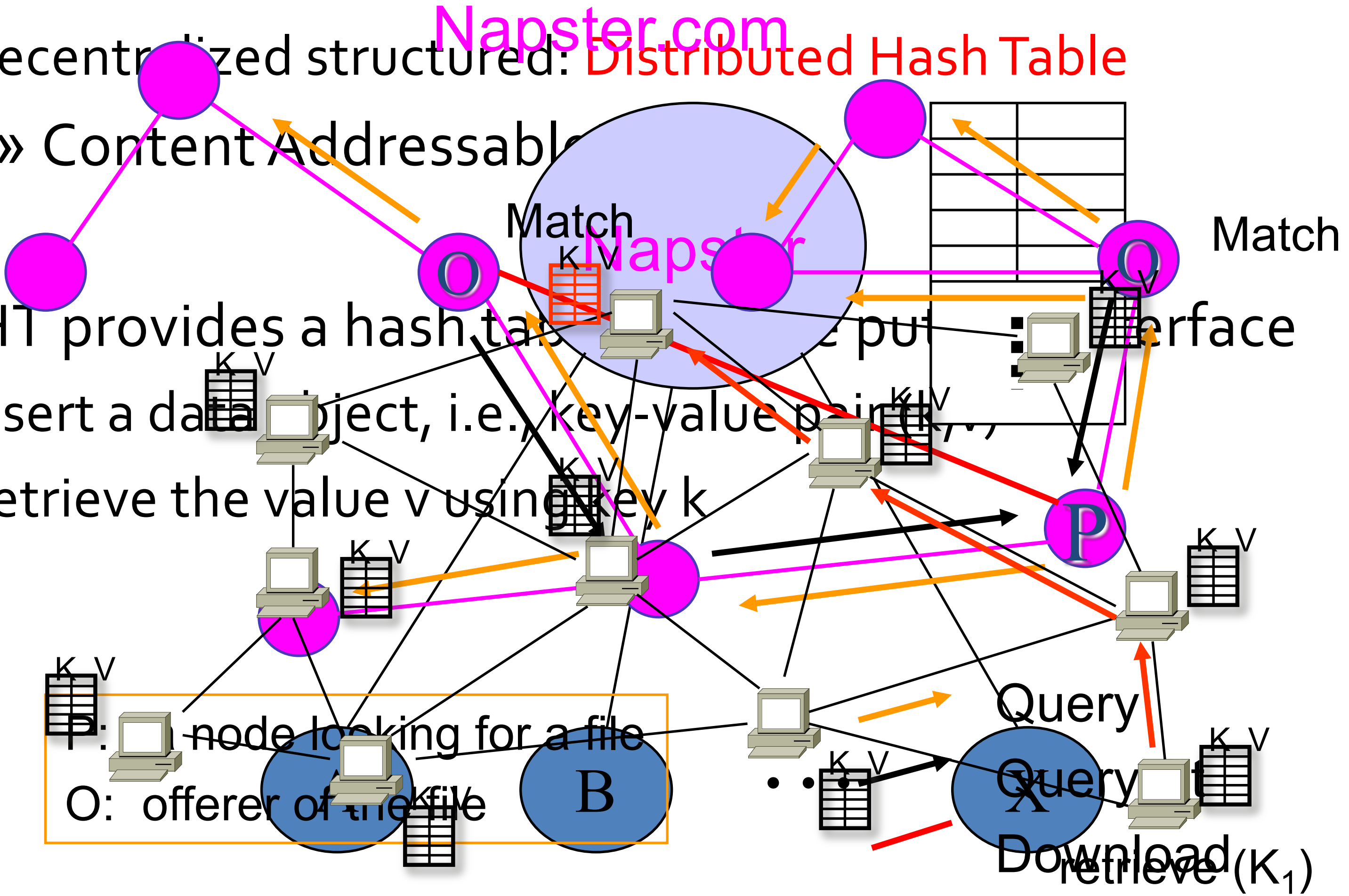
- File Sharing : Napster, Gnutella, BitTorrent, etc
- Commercial Applications
  - Blockchain
  - Skype
- Research community
  - P2P File and archival systems: Ivy, Kosha, Oceanstore, CFS
  - Web caching: Squirrel, Coral
  - Multicast systems: SCRIBE
  - P2P DNS: CoDNS and CoDoNS
  - Internet routing: RON
  - Next generation Internet Architecture: I3

Issues in P2P Systems

- Identity
  - Who am I talking to?
- Routing
  - How to find desired information?
- Trust
  - How do I know my peers behave nicely?
- Churn (Dynamicity)
  - Peers come and go.
- Incentivization
  - How to make peers to contribute to the system?

P2P Routing

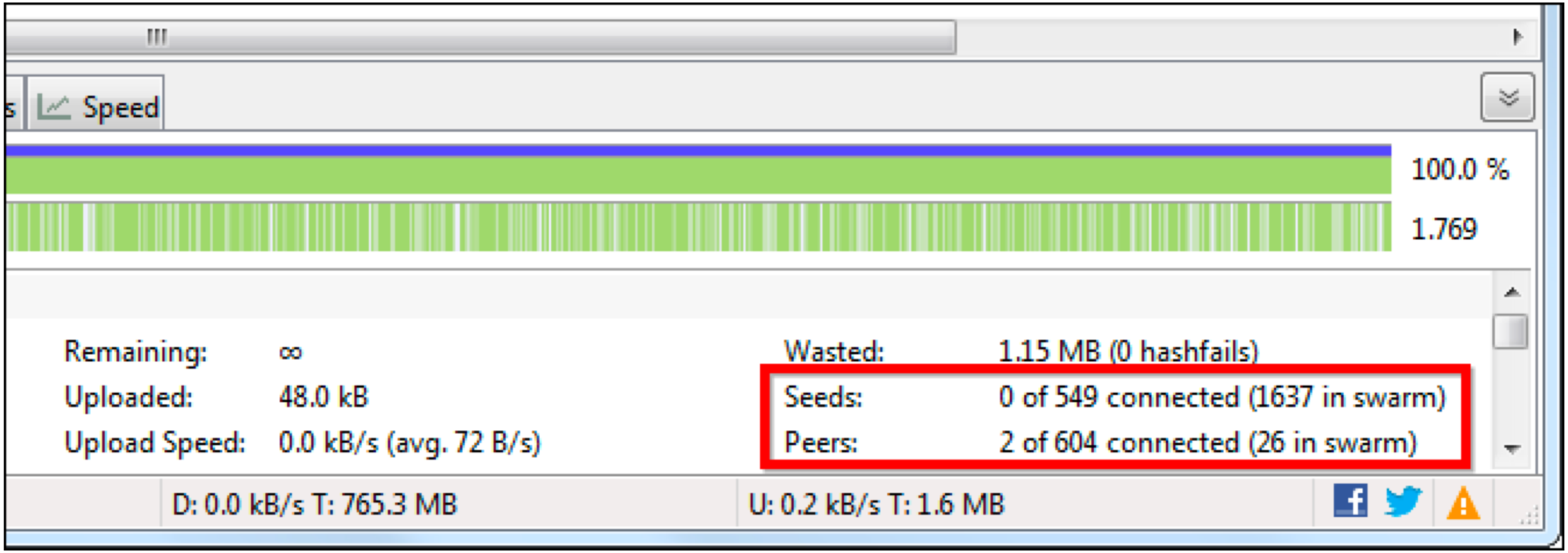
- How to find the desired information?
  - Centralized structured: Napster
  - Decentralized unstructured: Gnutella
  - Decentralized structured: Distributed Hash Table
    - » Content Addressabl[illegible]
- A DHT provides a hash ta[illegible] put[illegible]erface
  - Insert a data object, i.e., key-value pair (k,v)
  - Retrieve the value v using key k



Case Study: BitTorrent

- A computer joins a BitTorrent swarm by loading a .torrent file into a BitTorrent client.
- The client contacts a "tracker" specified in the .torrent file.
  - The tracker shares their IP addresses with other clients in the swarm, allowing them to connect to each other.
- Once connected, a client downloads bits of the files in the torrent in small pieces, downloading all the data it can get.
- Once the client has some data, it can then begin to upload that data to other BitTorrent clients in the swarm.
- In this way, everyone downloading a torrent is also uploading the same torrent.

Case Study: BitTorrent

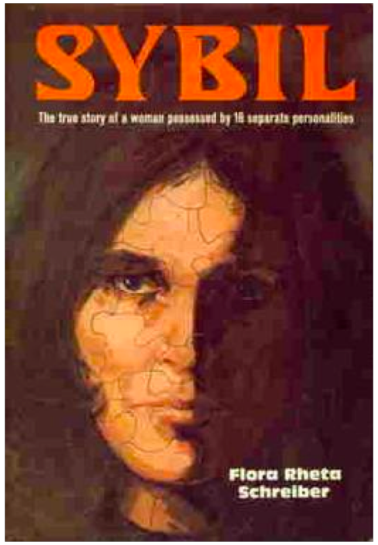


```
Remaining:      ∞                            Wasted:   1.15 MB (0 hashfails)
Uploaded:       48.0 kB                      Seeds:    0 of 549 connected (1637 in swarm)
Upload Speed:   0.0 kB/s (avg. 72 B/s)       Peers:    2 of 604 connected (26 in swarm)

D: 0.0 kB/s T: 765.3 MB        U: 0.2 kB/s T: 1.6 MB
```

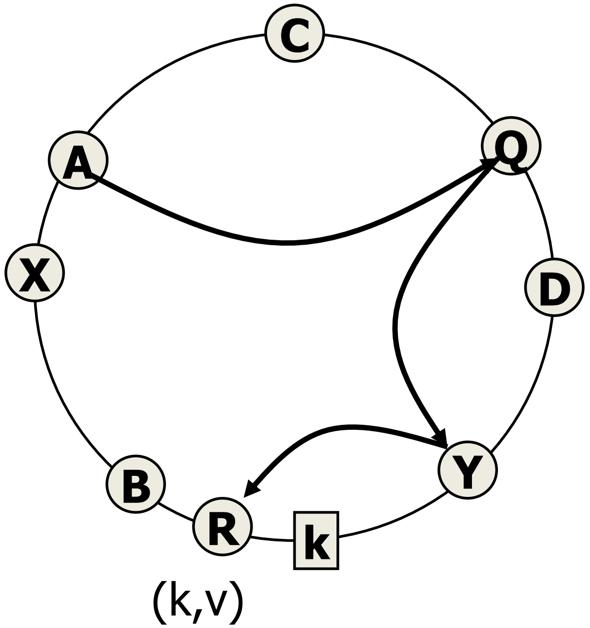
Attacks on P2P Systems

- ❑ Sybil Attack
  - ▹ the attacker subverts the reputation system of a P2P network by creating a large number of pseudonymous identities, to gain a large influence


SYBIL
The true story of a woman possessed by 16 separate personalities
Flora Rheta Schreiber

- ❑ Eclipse Attack (aka routing-table poisoning)
  - ▹ attacker takes over the peer's routing table so that they are unable to communicate with any other peer except the attacker

DHT: Terminologies



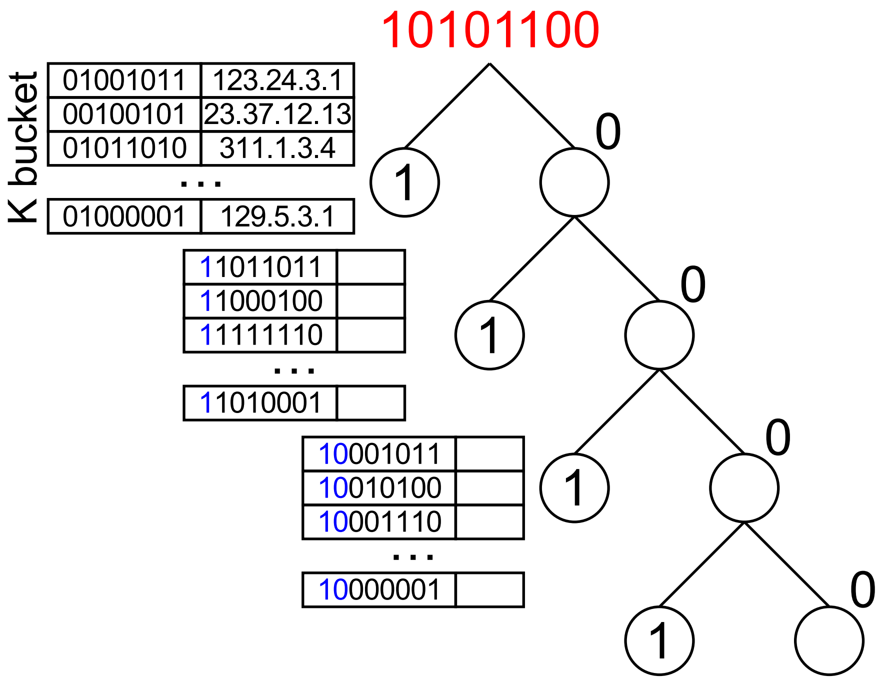
- ❑ Every node has a unique ID: *nodeID*
- ❑ Every object has a unique ID: *key*
- ❑ Keys and nodeIDs are logically arranged on a *ring* (*ID space*)
- ❑ A data object is stored at its *root(key)* and several *replica roots*
  - ▹ Closest nodeID to the key (or successor of k)
- ❑ *Range:* the set of keys that a node is responsible for
- ❑ Routing table size: O(log(N))
- ❑ Routing delay: O(log(N)) hops
- ❑ Content addressable!

Target P2P System

- ❑ Kad
  - ▹ A peer-to-peer DHT based on Kademlia

- ❑ Kad Network
  - ▹ Overnet: an overlay built on top of eDonkey clients
    - » Used by P2P Bots
  - ▹ Overlay built using eD2K series clients
    - » eMule, aMule, MLDonkey
    - » Over 1 million nodes, many more firewalled users
  - ▹ BT series clients
    - » Overlay on Azureus
    - » Overlay on Mainline and BitComet

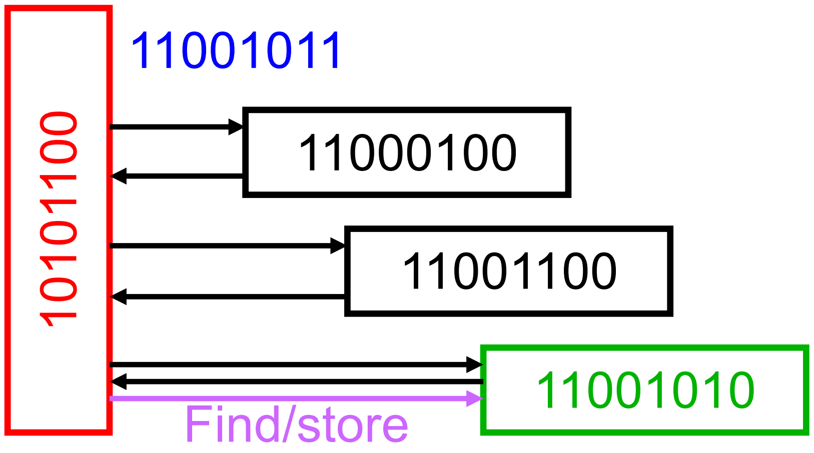
Kademlia Protocol

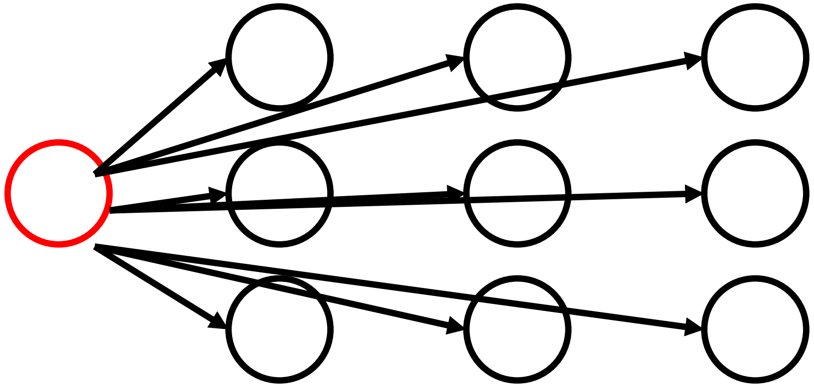


01001011	123.24.3.1
00100101	23.37.12.13
01011010	311.1.3.4
…	
01000001	129.5.3.1

11011011	
11000100	
11111110	
…	
11010001	

10001011	
10010100	
10001110	
…	
10000001	

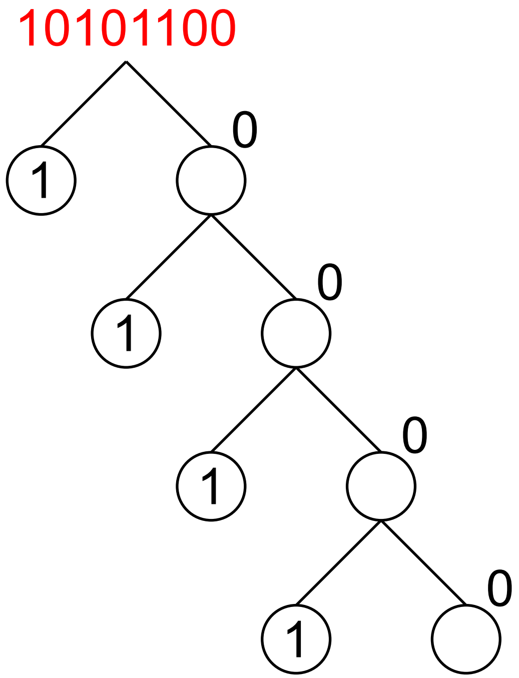


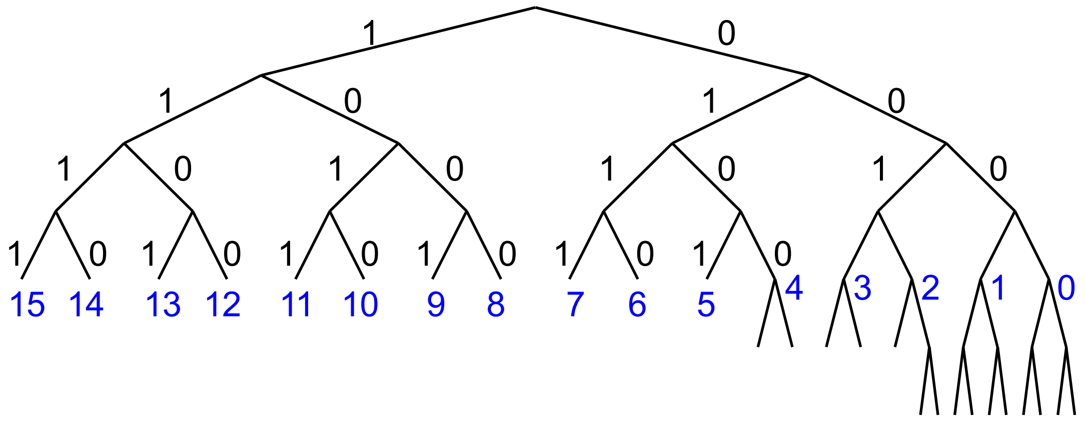


- d(X, Y) = X XOR Y
- An entry in k-bucket shares at least k-bit prefix with the nodeID
  - k=20 in overnet
- Add new contact if
  - k-bucket is not full

- Parallel, iterative, prefix-matching routing
- Replica roots: k closest nodes

Kad Protocol





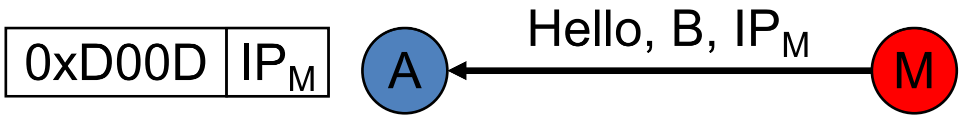
- ❑ No restriction on nodeID
- ❑ Replica root: |r, k| < δ
- ❑ K buckets with index [0,4] can be split if new contact is added to full bucket
- ❑ Wide routing table ➔ short routing path
- ❑ K bucket in i-th level covers $1/2^i$ ID space
- ❑ A knows new node by asking or contact from other nodes
- ❑ Hello_req is used for liveness
  - ▹ routing request can be used

Vulnerabilities of Kad

- No admission control, no verifiable binding
  - An attacker can launch a Sybil attack by generating an arbitrary number of IDs
- Eclipse Attack
  - Stay long enough: Kad prefers long-lived contact
  - (ID, IP) update: Kad client will update IP for a given ID without any verification
- Termination condition
  - Query terminates when A receives 300 matches.
- Timeout
  - When M returns many contacts close to K, A contacts only those nodes and timeouts.

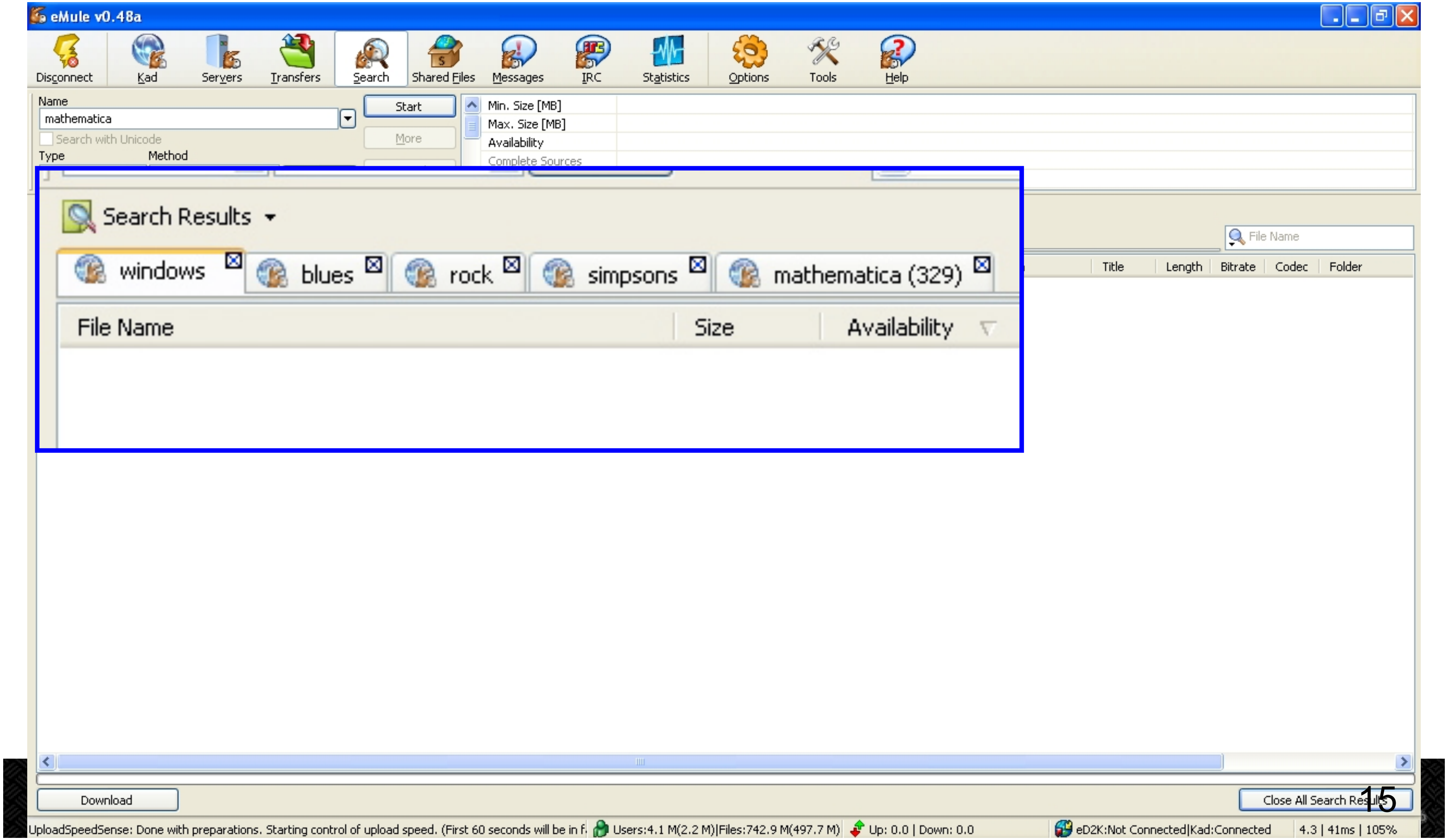
Actual Attack

- Preparation phase
  - Backpointer Hijacking: 8 A, attacker M
    - » Learns A's Routing Table by sending appropriate queries
    - » Then, change routing table by sending the following message.



- Execution phase
  - Provide many non-existing contacts
    - » Fact: Query will timeout after trying 25 contacts.

Screen Shots



eMule v0.48a

Name: mathematica

Search Results: windows, blues, rock, simpsons, mathematica (329)

UploadSpeedSense: Done with preparations. Starting control of upload speed. (First 60 seconds will be in f. Users:4.1 M(2.2 M)|Files:742.9 M(497.7 M) Up: 0.0 | Down: 0.0 eD2K:Not Connected|Kad:Connected 4.3 | 41ms | 105%

Summary of Estimated Cost

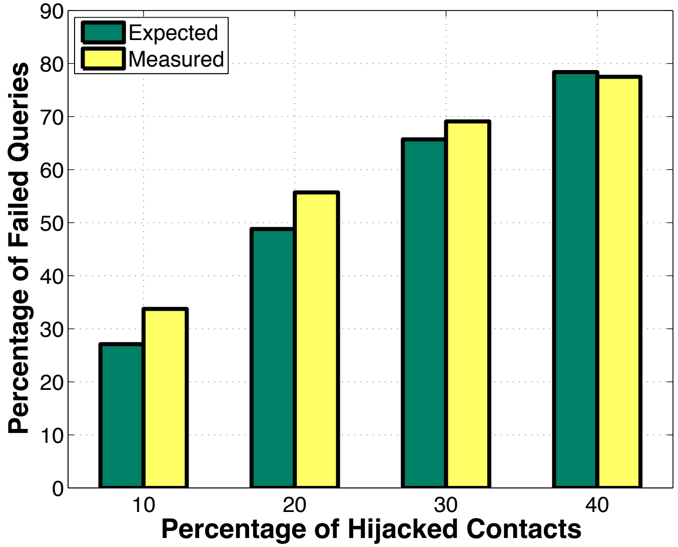
- ❑ Assumption
  - Total 1M nodes
  - 800 routing table entries
  - 100 Mbps network link

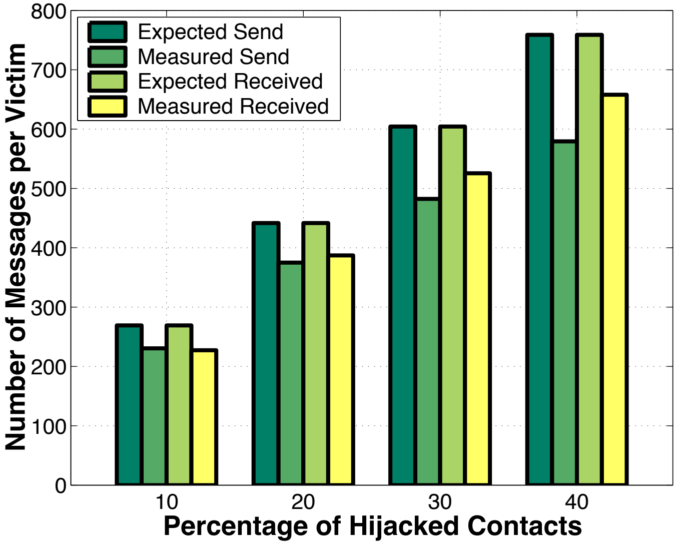
- ❑ Preparation cost
  - 41.2GB bandwidth to hijack 30% of routing table
  - Takes 55 minutes with 100 Mbps link

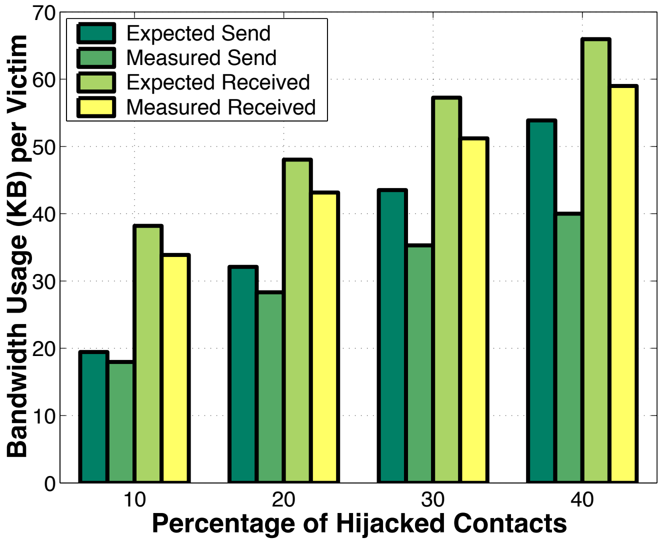
- ❑ Query prevention
  - 100 Mbps link is sufficient to stop 65% of WHOLE query messages.

Large scale simulation

- ❑ 11,303 ~ 16,105 Kad nodes running on ~500 PlanetLab machines





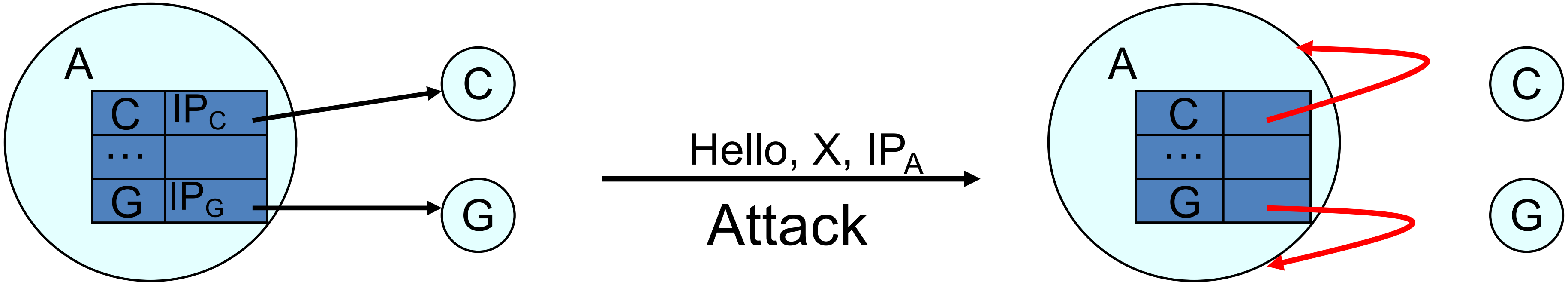


Comparison between expected and measured

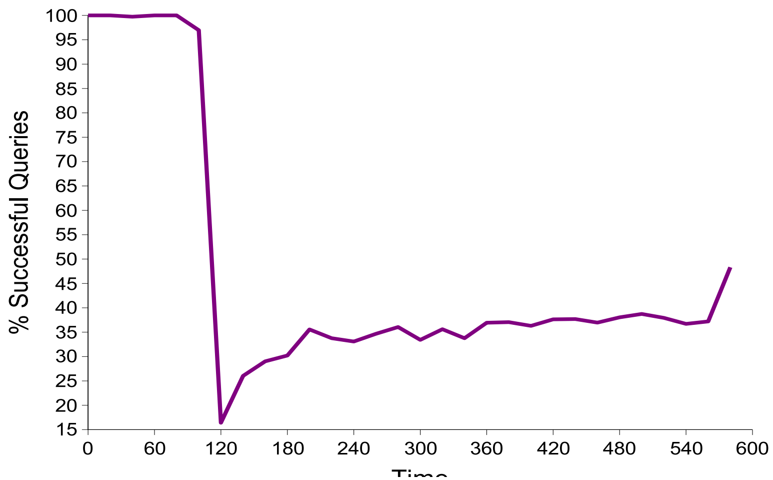
- keyword query failures
- Number of messages used to attack one node
- Bandwidth usage

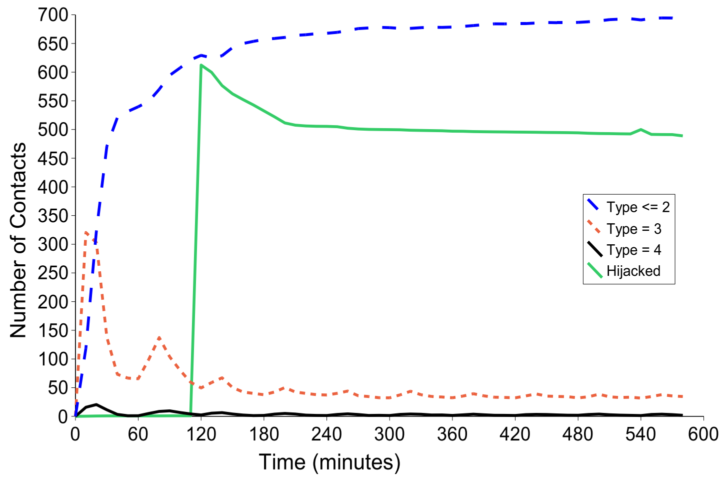
Self reflection attack

- Fill node A’s routing table with A itself.



- ≈ 100% queries failed after attack
- Nodes can recover slowly
- Second round of attack





Mitigations

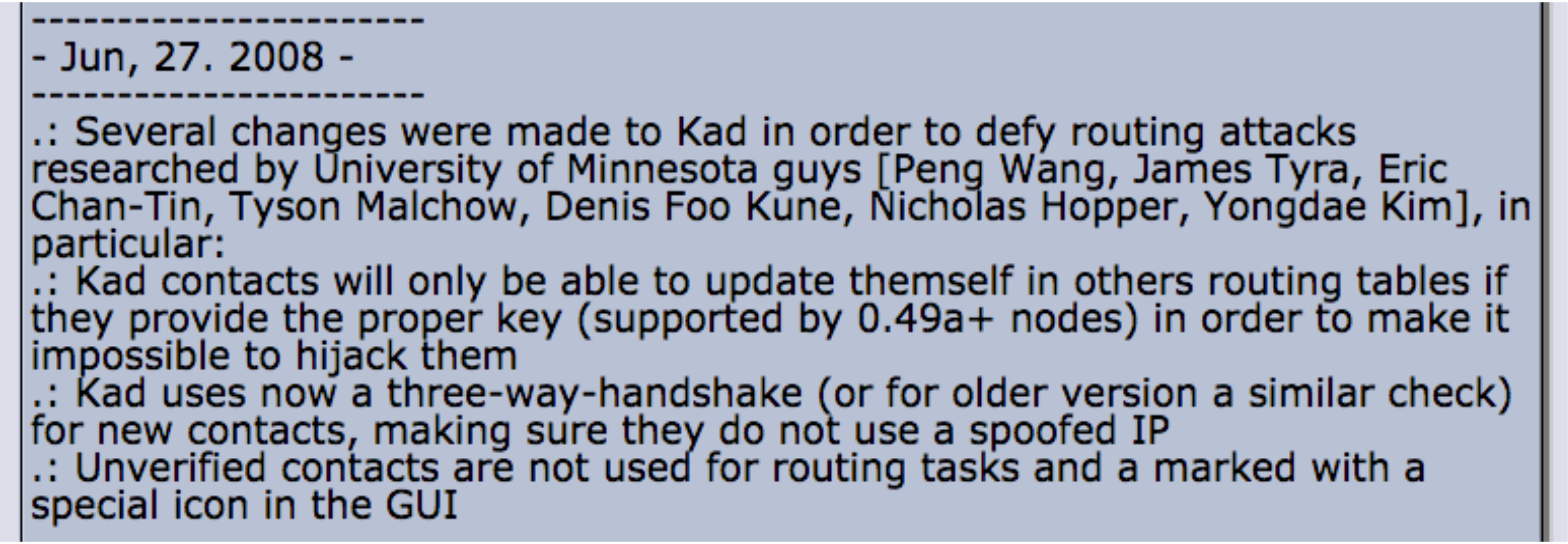
- Identity authentication

Method	Secure	Persistent ID	Incremental deployable
Verify the liveness of old IP	No	Yes	Yes
Drop Hello with new IP	Yes	No	Yes
ID=hash(IP)	Yes	No	No
ID=hash(Public Key)	Yes	Yes	No

- Routing correctness
  - Independent parallel routes
    - Incrementally deployable

bad pointers	Current method	Independent parallel routes
40%	98% fail	45% fail
10%	59.5% fail	1.7% fail

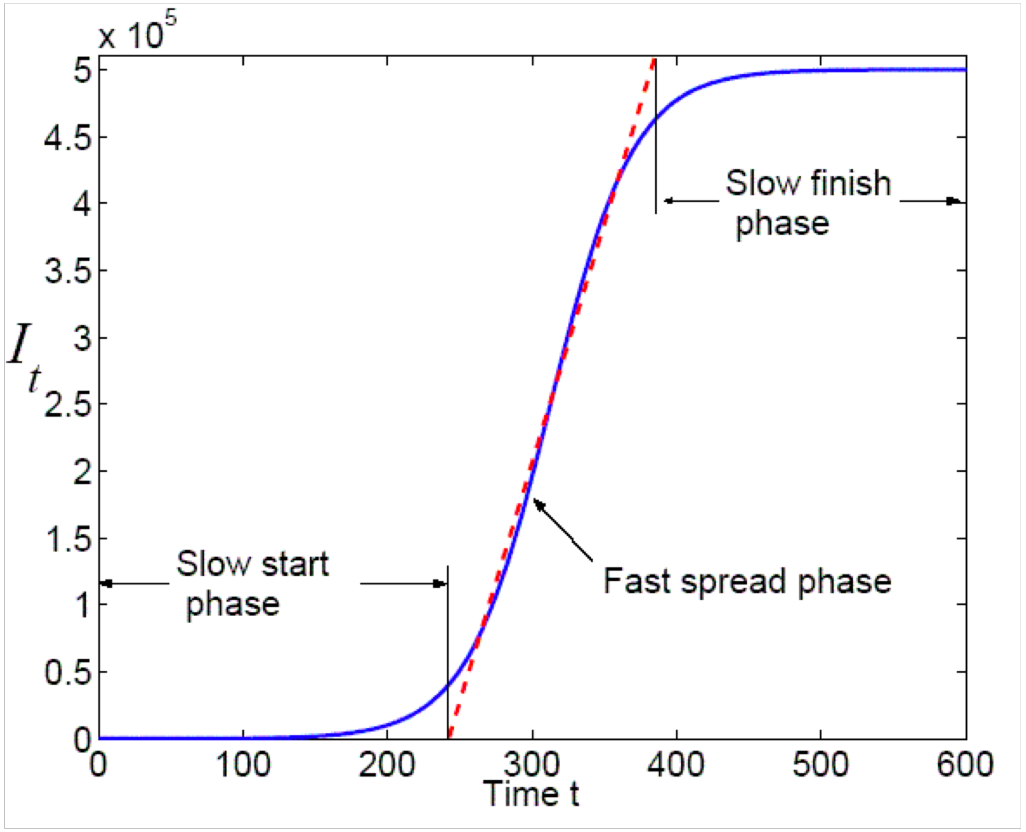
Then



```
------------------------
- Jun, 27. 2008 -
------------------------
.: Several changes were made to Kad in order to defy routing attacks
researched by University of Minnesota guys [Peng Wang, James Tyra, Eric
Chan-Tin, Tyson Malchow, Denis Foo Kune, Nicholas Hopper, Yongdae Kim], in
particular:
.: Kad contacts will only be able to update themself in others routing tables if
they provide the proper key (supported by 0.49a+ nodes) in order to make it
impossible to hijack them
.: Kad uses now a three-way-handshake (or for older version a similar check)
for new contacts, making sure they do not use a spoofed IP
.: Unverified contacts are not used for routing tasks and a marked with a
special icon in the GUI
```

Gossip Protocols

- ❑ a process of P2P communication that is based on the way that epidemics spread
- ❑ How to distribute information to all peers?



Issues in P2P Gossip protocols

- ❑ Reliability
  - ▹ All members receive the information
- ❑ Latency
  - ▹ The time needed to deliver a message to all members
- ❑ Bandwidth
  - ▹ Total bandwidth consumption
- ❑ Network/Node Dynamics
  - ▹ When network changes or nodes churn
- ❑ Robustness against Sybil/Eclipse attack
- ❑ Incentivization
  - ▹ Incentive to forward

Questions?

- Yongdae Kim
  - email: yongdaek@kaist.ac.kr
  - Home: http://syssec.kaist.ac.kr/~yongdaek
  - Facebook: https://www.facebook.com/y0ngdaek
  - Twitter: https://twitter.com/yongdaek
  - Google “Yongdae Kim”