HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

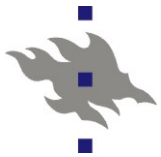
Overlay and P2P Networks

Summary

Samu Varjonen

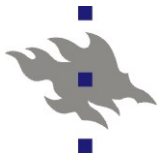
Ashwin Rao

29.2.2016



Contents

- Advanced topics - Security
- Revisiting Kademlia
- Summary



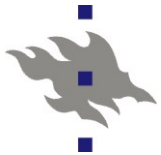
Key security issues

Security is a weak point of many overlays and DHTs

Not an issue with centrally managed system, but a
    significant concern for decentralized systems

Freenet was our key example of a secure P2P system
    Also Skype with centralized login servers

Overlays are vulnerable to the sybil attack
    One entity presents multiple identities for malicious intent.



Sybil attack

Sybil attack (John Doceur, IPTPS '01)

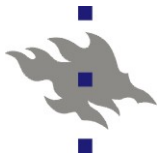
Problem: Malicious nodes overwhelm the network with identities

Resolution: Without a central authority that certifies identifies (binding real-world person to nodeID) no realistic approach exists to completely stop the sybil attack

With certain assumption sybil attacks can be mitigated

Geometric distinctness certification (Bazzi & Konjevod, PODC 2005)

Detect clock skew of the devices (Kohno, Broido, and claffy, UCSD, IEEE S&P 2005)



Eclipse attack

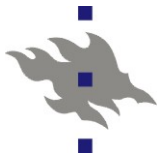
A group of malicious nodes tries to dominate the neighbor set

Start with Sybil and then neighbour discovery

- Adds malicious node identifiers to neighbour table
- For example: pick ids close to the target id

Result: network partitions

Solutions: Remember Freenet node identifier creation (the commitment protocol). Baseline: do not allow arbitary choice of node id.



Security Considerations

Malicious nodes

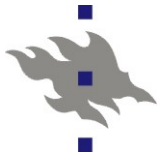
- Attacker floods DHT with data
- Attacker returns incorrect data
- Attacker denies data or supplies incorrect routing info

Basic solutions:

- rate limits and proof-of-work
- Self-authenticating data
- Redundancy
- **k-redundant networks**

What if attackers have quorum?

- Need a way to control creation of node Ids
- Solution: **secure node identifiers**
  - **Use public keys**



Security summary

Wide-area overlay networks are vulnerable to sybil and eclipse attacks

A production system typically utilizes centralized authentication

- Skype login servers

Decentralized systems

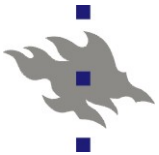
- Freenet: verification to protect content, darknet for better protection

If managed by one organization security becomes easier

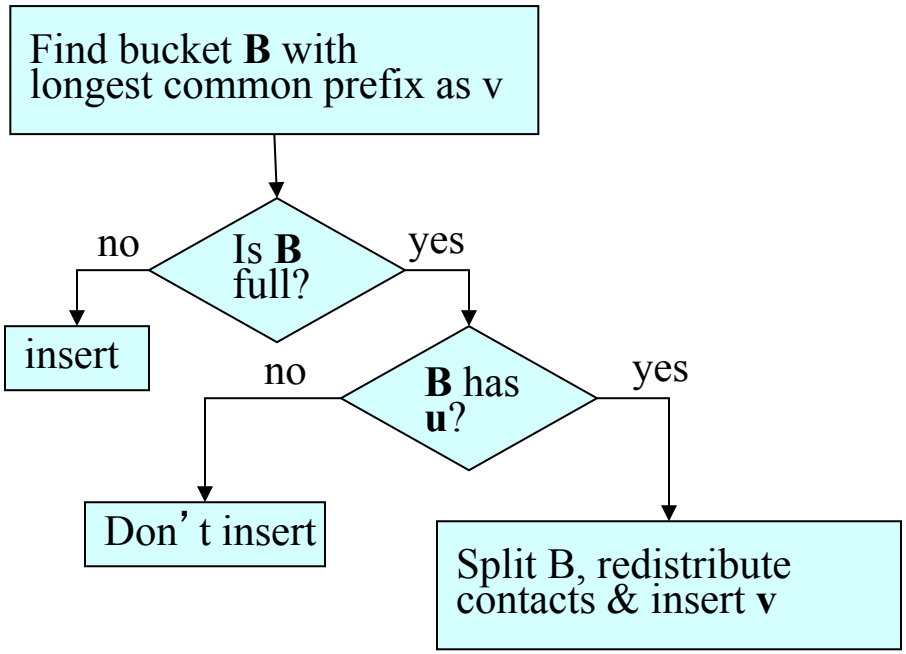
- Dynamo: no security considerations

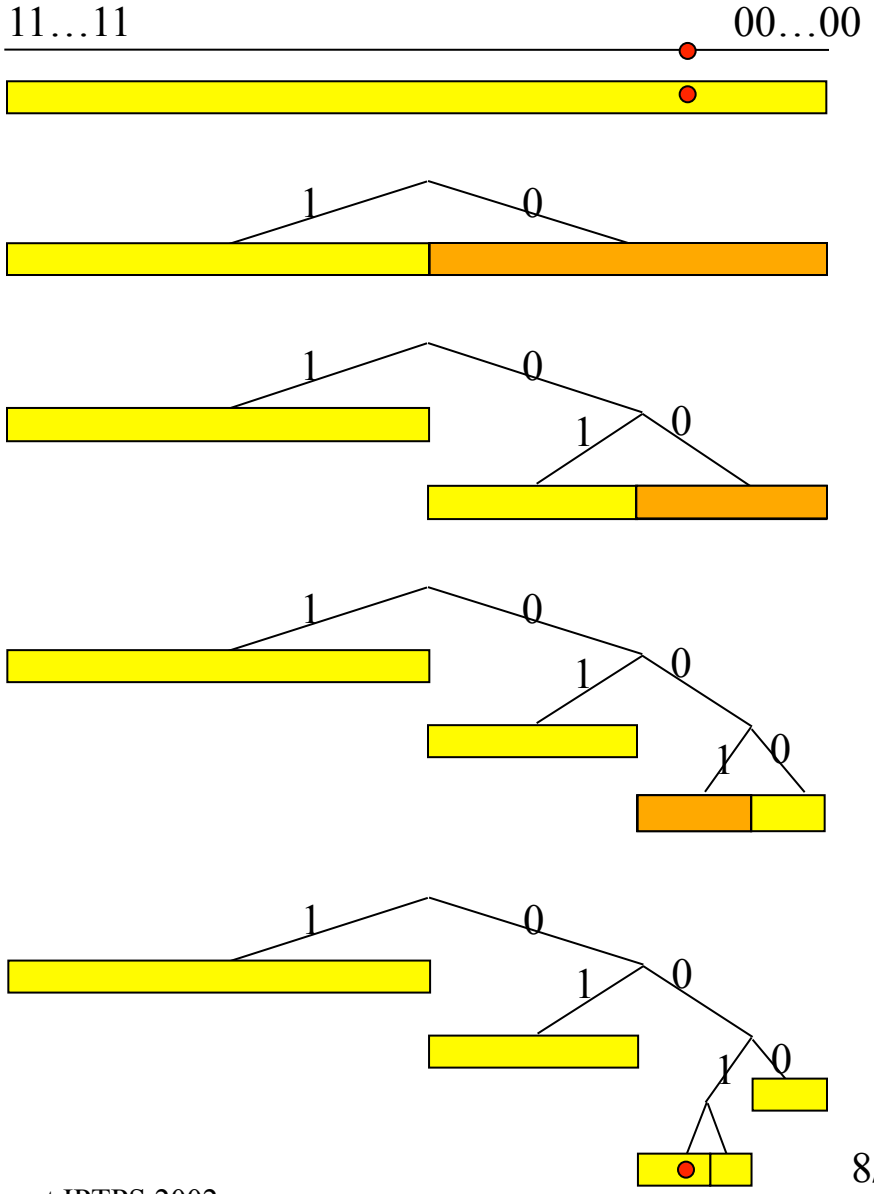
Re-visiting Kademlia

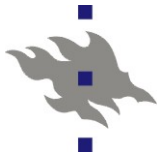
- XOR geometry
- Buckets
  - For each i ($0 \leq i < 160$) every node keeps a list (k-bucket) of nodes of distance between 2^i and $2^{(i+1)}$ from itself
  - The list is sorted by time last seen
  - The list is **updated** whenever a node receives a message
  - Nodes that are in the *n*th bucket must have a **differing *n*th bit from the node's identifier**



- Joining Node (**u**):
  - ✓ Borrow an alive node's ID (**w**) off-line
  - ✓ Initial routing table has a single k-bucket containing u and w.
  - ✓ u performs FIND_NODE(**u**) to learn about other nodes
- Inserting new entry (**v**)

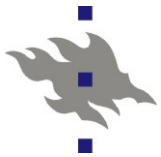






Kademlia

	Kademlia
Foundation	XOR metric
Routing function	Matching key and nodeID
System parameters	Number of peers N, base of peer identifier B
Routing performance	*$O(\log_B N)$ + small constant*
Routing state	*$B \log_B N + B$*
Joins/leaves	*$\log_B N$ + small constant*



DHT: A General Approach

What is an address?

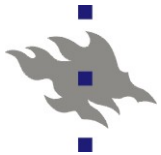
Base b with n digits

How to route efficiently?

Fix at least one digit per hop or take to the numerically closest destination based on routing table

How efficient is this?

Log N steps gives O(log N) state and O(log N) hops! Also we can observe m = log(N), and at most m hops fixing each digit so log(N) diameter



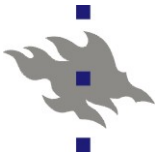
DHT: A General Approach

How to populate routing table?

Iterative nearest neighbour search to fill the routing table.
Get enough information to be able to populate the routing table.

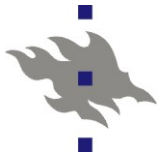
Unstructured or semi structured networks

	BitTorrent	Freenet v0.7	Gnutella v0.4	Gnutella v0.7
Decentralization	Centralized model	Similar to DHTs, two modes (darknet and opennet), two tiers	Flat topology (random graph), equal peers	Random graph with two tiers. Two kinds of nodes, regular and ulta nodes. Ultra nodes are connectivity hubs
Foundation	Tracker	Keywords and text strings are used to identify data objects. Assumes small world structure for efficiency	Flooding mechanism	Selective flooding using the ultra nodes
Routing function	Tracker	Clustering using node location and file identifier. Path folding optimization (opennet). Location swapping (darknet).	Flooding mechanism	Selective flooding mechanism
Routing performance	Guarantee to locate data, good performance for popular data	Search based on Hop-To-Live, no guarantee to locate data. With small world property $O(\log(n)^2)$ hops are required, where n is the number of nodes.	Search until Time-To-Live expires, no guarantee to locate data	Search until Time-To-Live expires, second tier improves efficiency, no guarantee to locate data
Routing state	Constant, choking may occur	With small world property $O(\log(n)^2)$	Constant (reverse path state, max rate and TTL determine max state)	Constant (regular to ultra, ultra to ultra). Ultra nodes have to manage leaf node state.
Reliability	Tracker keeps track of the peers and pieces	No central point of failure	Performance degrades when the number of peer grows. No central point.	Performance degrades when the number of peer grows. Hubs are central points that can be taken out.



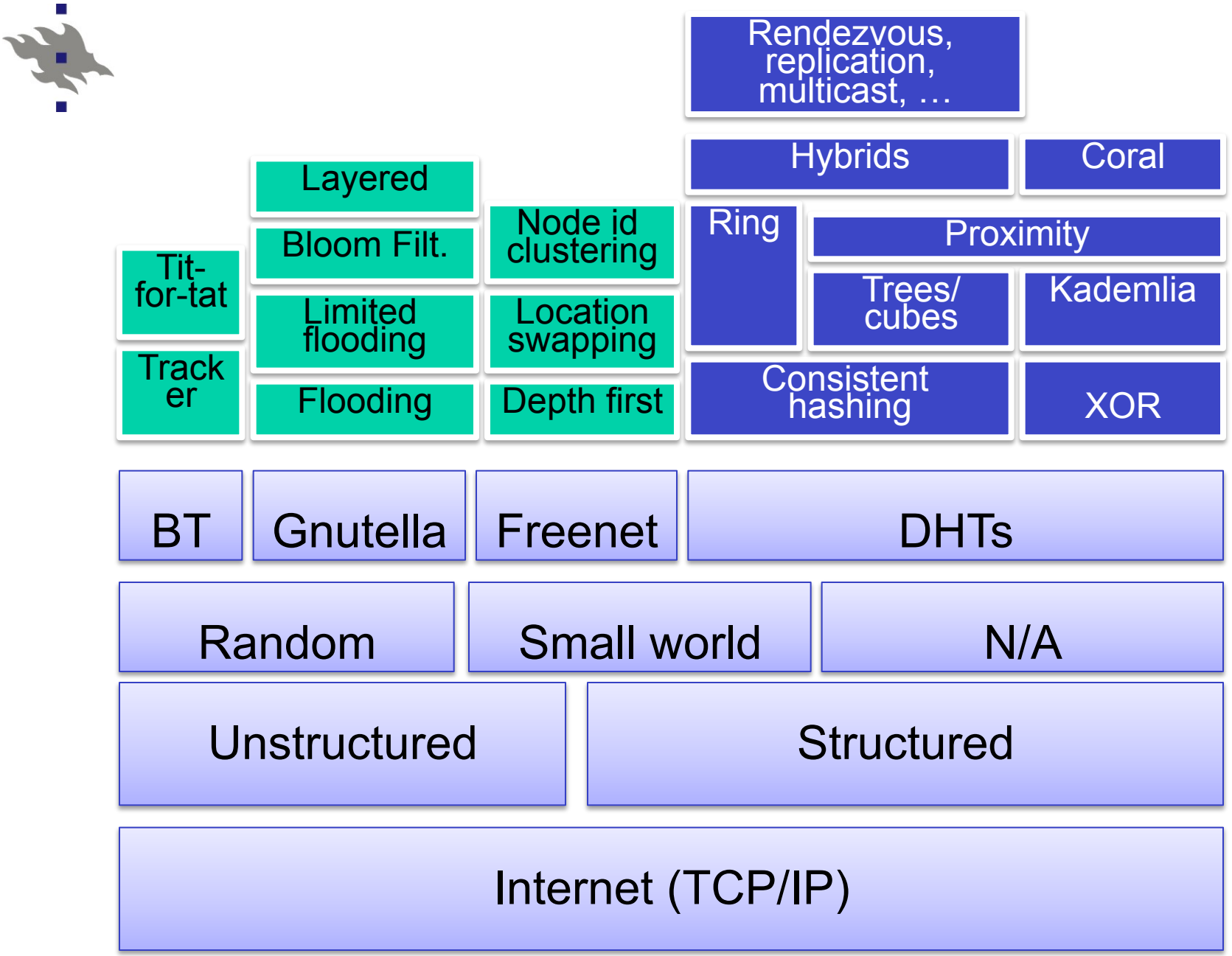
Structured networks

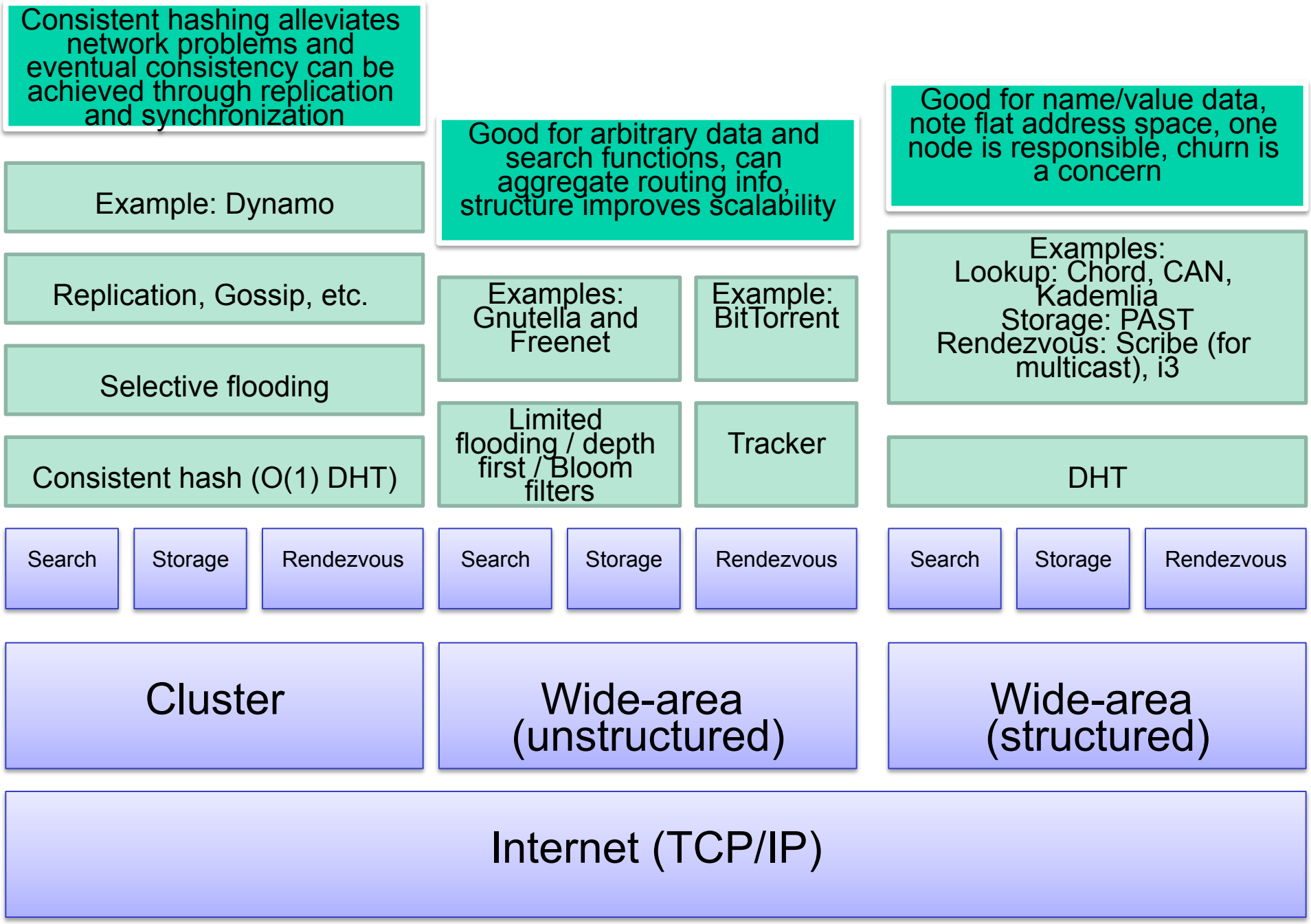
	CAN	Chord	Kademlia	Pastry	Tapestry
Foundation	Multi-dimensional space (d-dimensional torus)	Circular space	XOR metric	Plaxton-style mesh	Plaxton-style mesh
Routing function	Maps (key,value) pairs to coordinate space	Matching key and nodeID	Matching key and nodeID	Matching key and prefix in nodeID	Suffix matching
System parameters	Number of peers N, number of dimensions d	Number of peers N	Number of peers N, base of peer identifier B	Number of peers N, base of peer identifier B	Number of peers N, base of peer identifier B
Routing performance	$O(dN^{1/d})$	$O(\log N)$	$O(\log_B N)$ + *small constant*	$O(\log_B N)$	$O(\log_B N)$
Routing state	$2d$	$\log N$	$B\log_B N + B$	$2B\log_B N$	$\log_B N$
Joins/leaves	$2d$	$(\log N)^2$	$\log_B N$ + *small constant*	$\log_B N$	$\log_B N$



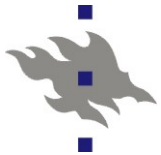
Summary of Datastores and Related Systems

Name	Type	Data placement	Data can be located	Replication	Stabilization
Dynamo	1-hop DHT	Key→node, consistent hashing	Yes	Yes, clockwise nodes	Eventual consistency: anti-entropy, gossip
Chord	Wide-area DHT	Key→node, based on consistent hashing	Yes	Add-on	Stabilization mechanism
PAST	Wide-area DHT	Key→node, based on Plaxton mesh	Yes	Yes, close-by nodes	Part of Pastry routing
Freenet	P2P	Key→closest node, approximate small world routing table	No guarantee	Yes, reverse-path	Path folding (opennet) and location swapping (darknet)
Gnutella	P2P	Client	No guarantee	No	No (ultranode layer in principle)


Rendezvous, replication, multicast, …
Hybrids
Coral
Layered
Node id clustering
Ring
Proximity
Tit-for-tat
Bloom Filt.
Trees/ cubes
Kademlia
Limited flooding
Location swapping
Track er
Flooding
Depth first
Consistent hashing
XOR
BT
Gnutella
Freenet
DHTs
Random
Small world
N/A
Unstructured
Structured
Internet (TCP/IP)


Consistent hashing alleviates network problems and eventual consistency can be achieved through replication and synchronization
Example: Dynamo
Replication, Gossip, etc.
Selective flooding
Consistent hash (O(1) DHT)
Search
Storage
Rendezvous
Cluster
Good for arbitrary data and search functions, can aggregate routing info, structure improves scalability
Examples: Gnutella and Freenet
Example: BitTorrent
Limited flooding / depth first / Bloom filters
Tracker
Search
Storage
Rendezvous
Wide-area (unstructured)
Good for name/value data, note flat address space, one node is responsible, churn is a concern
Examples: Lookup: Chord, CAN, Kademlia Storage: PAST Rendezvous: Scribe (for multicast), i3
DHT
Search
Storage
Rendezvous
Wide-area (structured)
Internet (TCP/IP)

Main theme	Prerequisites	Approaches learning goals	Meets learning goals	Deepens learning goals
Overlay and peer-to-peer networks: definitions and systems	Basics of data communications and distributed systems (Introduction to Data Communications, Distributed Systems)	Knowledge of how to define the concepts of overlay and peer-to-peer networks, and state their central features Ability to describe at least one system in detail	Ability of being able to compare different overlay and p2p networks in a qualitative manner Ability to assess the suitability of different systems to different use cases	Ability to give one's own definition of the central concepts and discuss the key design and deployment issues
Distributed hash tables	Basics of data communications and distributed systems (Introduction to Data Communications, Distributed Systems) Big-O-notation and basics of algorithmic complexity	Knowledge of the concepts of structured and unstructured networks and the ability to classify solutions into these two categories Knowledge of the basics of distributed hash tables Ability to describe at least one distributed hash table algorithm in detail	Ability of being able to compare different distributed hash table algorithms Ability of designing distributed hash table-based applications Knowledge of key performance issues of distributed hash table systems and the ability to analyze these systems	The knowledge of choosing a suitable distributed hash table design for a problem Familiarity with the state of the art
Reliability and performance modelling	Basics of probability theory Basics of reliability in distributed systems	Ability to model and assess the reliability of overlay and peer-to-peer networks by using probability theory Knowledge of the most important factors pertaining to reliability	Ability of analytically analyzing the reliability and performance of overlay and peer-to-peer networks Understanding of the design issues that are pertinent for reliable systems	Familiarity with the state of the art
Content distribution	Introduction to Data Communications	Knowledge of the basic content distribution solutions Ability to describe at least one overlay and p2p network based content distribution solution	Knowledge of different content distribution systes and the ability to compare them in detail Knowledge of several content distribution techniques	Familiarity with the state of the art
Security	Basics of computer security	Knowledge of the basic security issues with overlay and p2p networks Knowledge of the sybil attack concept	Ability to discuss how security problems and limitations can be solved Knowledge of how to prevent sybil attacks	Knowledge of how to prevent sybil attacks Familiarity with the state of the art



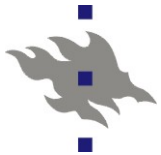
Course Development

General feedback

More hands-on assignments?

- Separate programming task for more credits?

MOOC based questions?



Have a nice Spring!

Remember course feedback