Freenet: Clarke, Miller, Hong, Sandberg, Wiley [Clarke00a]

CSci551: Computer Networks SP2006 Thursday Section John Heidemann

15b_Clarke02a: CSci551 SP2006 © John Heidemann

Peer-to-peer Systems Intro

• why p2p?

- many reasons
- Freenet: anonymity
- Napster: how to easily find content
- Bittorrent: exploiting parallelism in download process
- Chord: binary search around ring
- things to look for in p2p systems
 - search (both *name to key* and *key to location*), update, redundancy/fault-tolerance

4

10

12

15b_Clarke02a: CSci551 SP2006 © John Heidemann

Key ideas

- · distributed storage system
- anonymity
 - search/routing indirectly through other peers
 - data can be stored anywhere
 - · difficult to censor or stop data in the net • no records of who posted data: difficult to find poster
 - keep data encrypted
 - · you can't tell what data you have
- routing via "node chains"
- and optimizing routing via "hill climbing" approach • replicate files on retrieval (to improve fault
- tolerance and performance)

```
15b_Clarke02a: CSci551 SP2006 © John Heidemann
```

Preliminaries: the Politics

- Freenet (more than other protocols) has an explicit political goal
 - distribute data (any data)
 - anonymously
- builds on prior work in anonymous email

15b_Clarke02a: CSci551 SP2006 © John Heidemann

Freenet Components: GUIDs

• GUIDs

» globally unique identifiers: SHA-1 hashes of something, used to locate the key

– goal: have short identifier to stand in for filename or keyword or contents

15b_Clarke02a: CSci551 SP2006 © John Heidemann

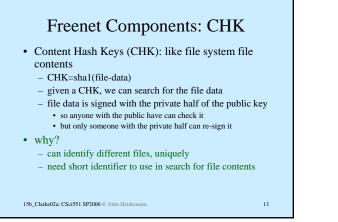
11

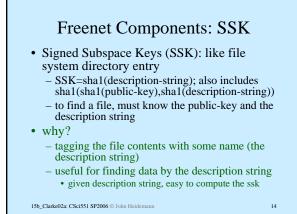
1

Freenet Components: public key pairs • files have a *contents*, and a *public-key pair*, and one or more *description strings* content is Declaration of Independence, key xxx, string: politics/doi store the SSK under h(politics/doi), that then points to h(contents) store the CHK under h(contents) • why? "allow users to create their own space" - need some mechanisms to allow someone to make changes to their data · vet don't want to know who they are authenticate to some anonymous user X, not a name like "John Smith"

- compare to other systems which add a PKI: public key infrastructure that maps public keys through a chain of trust back to known third party

15b_Clarke02a: CSci551 SP2006 © John Heidemann





Basic Idea: Finding Data

• routing (not a standard term for this)

- generate a key (SSK) from the filename
 - it's just a hash, a "random" 160-bit number
 - gives the CHK
- find data by looking up CHK in network
- · search/discovery
 - finding the filename in the first place
 - out-of-band, or maybe do automatic indexing, or sharing names public

15b_Clarke02a: CSci551 SP2006 © John Heidemann

Basic Idea: Routing

- · throw data into a mesh of nodes
- each node has a routing table listing which neighbors have which keys
- · route queries towards keys
- encourage locality in where the keys are stored; how: replicate data as it is returned to the user
- performance
 - worst case: O(n) where n is the number of nodes
 - work case. Our matter in white in market is market of the strong answer, but their simulations say with 10k nodes, only 8 hop search most of the time locality: hope is that hash values promote locatility around similar

20

15b_Clarke02a: CSci551 SP2006 © John Heidemann

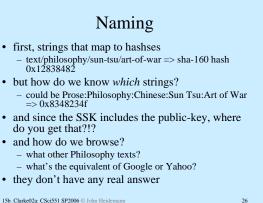
Basic Idea: Anonymity

- when propagating requests, add randomness to obscure sender/receiver
 - examples: each node pretends to be the original requests, nodes can tweak TTLs
 - where have we seen this before? xxx
- data is encrypted and stored by key, so node owner doesn't know contents
- updates are hard use public-key encryption to allow only owner to update

15b_Clarke02a: CSci551 SP2006 © John Heidemann

25

16



15b_Clarke02a: CSci551 SP2006 © John Heidemann

Updating

- how do we update data in place?
 - can't just replace data, because that allows denial of service
 - yet need to update data in place (ex. to maintain directories of keys)

• single user

- can use public key cryptography and indirection
- encryption/security details: see CSci555

28

32

15b_Clarke02a: CSci551 SP2006 © John Heidemann

Compare to Other Peer Systems

- Napster had a central database, it's distributed
 - Kazaa and Morpheous too (right?)
- Gnutella?
- others have better
 - search
 - user interface?

15b_Clarke02a: CSci551 SP2006 © John Heidemann

Does it work?

- Not clear if Freenet scales...
- with sparse keyspace, how much flooding?
 Vulnerable to DoS attacks...
 record companies putting songs with 15s of
 - music and then a raspberry – no real way to stop this; why?
- xxx • Not clear that search is sufficient..
- But very interesting design point

15b_Clarke02a: CSci551 SP2006 © John Heidemann

Comparing to Other p2p Systems: FreeNet

29

37

- search:
 - finding a name: no real directory system
 finding a key: hill climing algorithm
- update:
 - insert: just like search
 - update in place: uses public key stuff
- redundancy: many copies of file X
- other features: anonymity

15b_Clarke02a: CSci551 SP2006 © John Heidemann

