Computer Networks
ICS 651

- Peer-to-peer (P2P) networks
- bittorrent
- Distributed Hash Tables, DHTs
- bitcoin
- allnet

Peer-to-Peer (P2P) networks

- moving away from client-server model:
  - everyone has both client and server functionality
    - if using TCP, still has accept/connect distinction
  - nobody is authoritative
- as decentralized as possible
- logically every participant is both client and server, can both connect and be connected to
- often special-purpose, still no general-purpose P2P network
- usually **overlay networks** built on top of the existing Internet protocols

BitTorrent

- P2P content delivery/distribution network for files
- files may be large: bittorrent supports requesting and delivering parts of files
- each part of a file is identified by a hash
- torrent file lists all the hashes for a file
- a peer that has downloaded a part of a file can forward it to other peers
- a **tracker node** keeps track of peers that have parts of files

BitTorrent Reliability

- initial distribution (seeding) must succeed for content to be available
- for downloads to be possible, at least one tracker must be available to peers
- popular files are downloaded from several peers at once, maximizing bandwidth and minimizing the load on any one peer
- two peers can exchange different parts of the same file
  - peers can preferentially send content to peers from which they receive content
- unpopular files may disappear and no longer be available

Distributed Hash Table: Goals

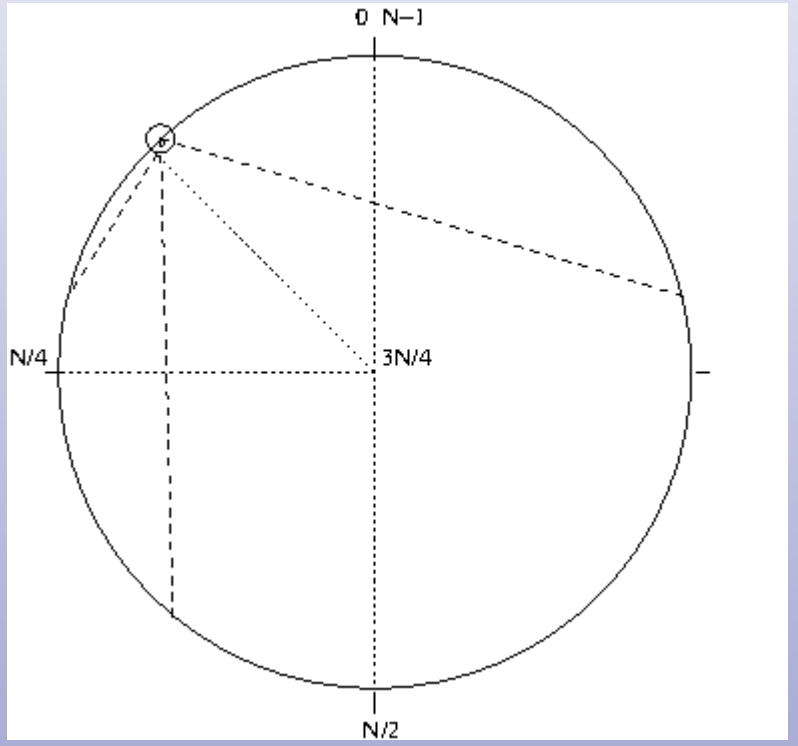
- a hash table is a data structure that uses randomness to distribute content uniformly across a table
- a Distributed Hash Table (DHT) uses randomness to distribute content uniformly across many machines
  - and reliably provide access to that content even in the face of machines joining and leaving the network (**churn**)
- a DHT is a P2P self-organizing storage network – and sometimes also a communications network
- access to one node of the network should give access to the entire content of the network
- data is identified by keys which are fixed-length bitstrings
  - same as in a hash table

DHT ID space and routing

- each node selects a random k-bit ID
- each node keeps a "routing table" with up to k entries
- the i-th entry in the routing table, if any, has a route to a node that has the same first i-1 bits of the ID, and a different i-th bit
- in this way, the first entry in the routing table is in the other half of the ID space
- the second entry in the routing table is in the same half of the ID space, but the other quarter, and so on
- to route a packet from node x to ID y, where the first i bits of x and y are the same, the packet is sent to the node (if any) in the ith position of the routing table -- this node will be closer to the destination
- if the network has N nodes, the routing table only needs log N entries

DHT ID space

- the first entry in the routing table is in the other half of the ID space
- the second entry in the routing table is in the same half of the ID space, but the other quarter
- the third entry in the routing table is in the same 1/4 of the ID space, but the other 1/8



DHT storage

- each node stores data for its own ID, and all IDs up to n-1 nodes later in ID space (maybe n=4)
- a request is routed until there is no route to a closer node. At this point, the requested data should be on the node
- when nodes leave the network (churn), the data should still be available redundantly in other nodes -- a data redistribution scheme then tries to have every item in at least n nodes

DHT performance

- at least one node must be reachable in order to connect with the network
- messages for new data must go through an average of log N hops, where each hop may cross the entire Internet
- when there is a lot of churn, routing tables must be recomputed frequently and data stored in the network must be redistributed frequently

Bitcoin

- distributed, reliable ("hard to falsify") time-stamping network
- each time-stamp record ("block") includes new transactions, and a summary (hash from Merkle tree) of all previous transactions
- finding valid blocks require compute power:
  - first n bits of hash must all be zero
  - can only be found at random (we think!)
  - n evolves to keep the number of blocks generated at a near constant rate
- "block miners" get a reward for finding a block with a suitable hash
- each block includes all transactions since the last block
- P2P network distributes the blocks to all peers

Bitcoin Transactions

- each transaction has a list of input transactions, adding up to an amount B
- each transaction has a list of output amounts and public keys, adding up to an amount B' <= B
- if B' < B, the difference B - B' belongs to the block miner as a transaction fee
- the input transactions provide evidence of having a private key equivalent to the public key of the corresponding output transaction

Bitcoin block chain

- each bitcoin miner produces the new block and hashes it as many times as possible until it finds a hash with the first n bits zero
  - each block includes a time stamp and other fields that can be modified so that the block hashes to a different value each time
- the miner then broadcasts the block as fast as possible
- every other miner includes the hash of this block into a new block they try to generate
- two (or more) miners may find blocks A and B approximately simultaneously, and broadcast both throughout the network
- every miner has a choice of which block to include in the block they are working on
- the next successful block C will have picked one of the winners of the previous round, A or B
- almost all miners will now build their block on C, confirming either A or B, and one of them will produce a new block D
- eventually, only one of A or B is confirmed by the growing block chain

Bitcoin block chain weakness

- if I control almost all the hashing power in the network
- I compute two blocks A, B with valid hashes
  - block A includes my transfer of bitcoins to someone I am trying to cheat
  - block B does not include that transaction. Instead, it includes a different transaction giving those same inputs to somebody else
  - I distribute block A widely, keep B to myself
- other miners build on block A
- meanwhile, I build new blocks building on block B
- once the recipient of my transaction has accepted it, I distribute all the blocks I built on B, which form a longer chain than the chain built on A
- in the future miners build on the longest chain, which includes B and not A

Types of Blockchains

- The bitcoin blockchain is controlled by the miners: whoever has hash power wins
- the bitcoin blockchain is open to everyone
- change either of these assumptions, and you get a different blockchain:
  - still immutable, as long as the participants behave
- a **permissioned blockchain** is only open to selected participants
- a **private blockchain** is controlled by whoever set up the blockchain
- either of these may (or may not) be readable by the general public

Other applications of Blockchains

- distributed, reliable network for time-stamping transactions
- might be useful anywhere distributed recordkeeping is needed
  - e.g. real-estate transactions
- but proof-of-work is expensive!
  - proof-of-stake may be a less energy-intensive alternative
  - bankers like the idea of a distributed network limited to only carefully-selected participants
- in any case, there must be incentive for the miners
  - e.g. in bitcoin, miners get newly-minted coins and transaction fees

AllNet

- a distributed, P2P network designed to be useful even without Internet access
  - when the Internet is accessible, forms a DHT to store and forward message
- in a local-broadcast network, each node sees all messages, receives the ones it can decrypt
- chat application is low-bandwidth, so low performance is not an issue
- real issue: difficulty accessing P2P networks from mobile devices
  - WiFi ad-hoc mode is not widely accessible without rooting the device
  - Bluetooth and Bluetooth Low Energy are not as long range, and are not really designed for two-way p2p communication of arbitrary data
  - mobile devices are designed to save energy, so suspend long-running processes
- http://alnt.org/

AllNet status, 2020

- ad-hoc communication between Linux systems of ad-hoc 802.11 WiFi
- DHT used for communication over the Internet
- chat client, time service
- trace mechanism for debugging, similar to ping+traceroute

```
b1.01/16     0 hop
b3.00/16     1 hop          about 50m away!
b4.00/16     2 hop
b5.00/16     3 hop          2.519748s rtt
```