BitTorrent

Katrina LaCurts
MIT CSAIL
katrina@csail.mit.edu

BitTorrent Overview

peer

BitTorrent Overview

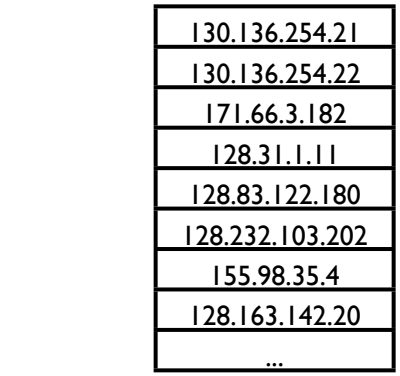
peer

tracker

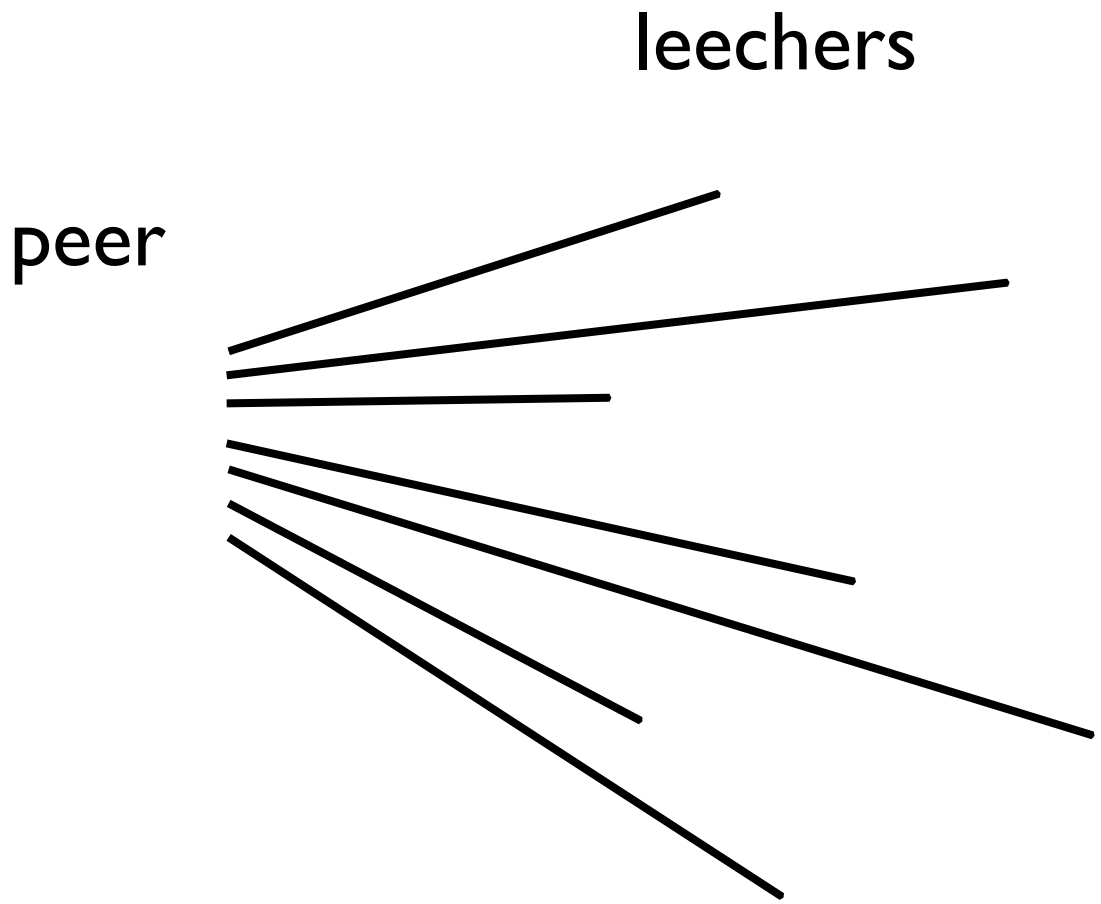
BitTorrent Overview

peer

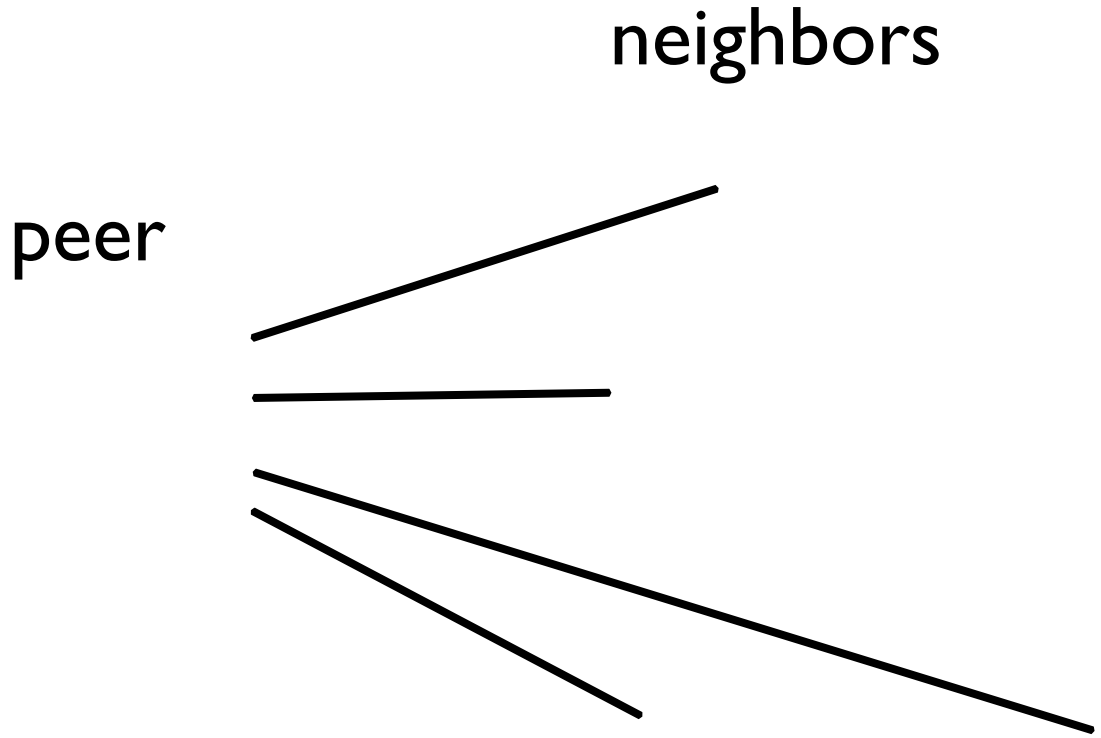
tracker



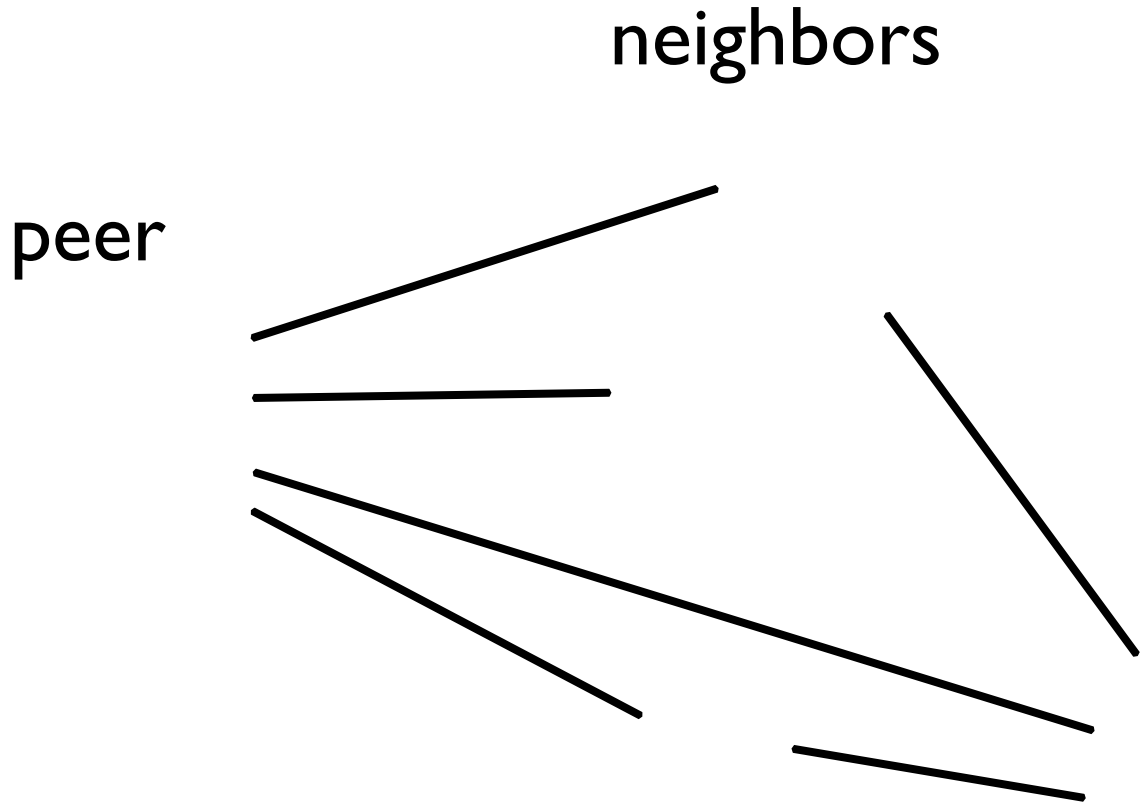
BitTorrent Overview



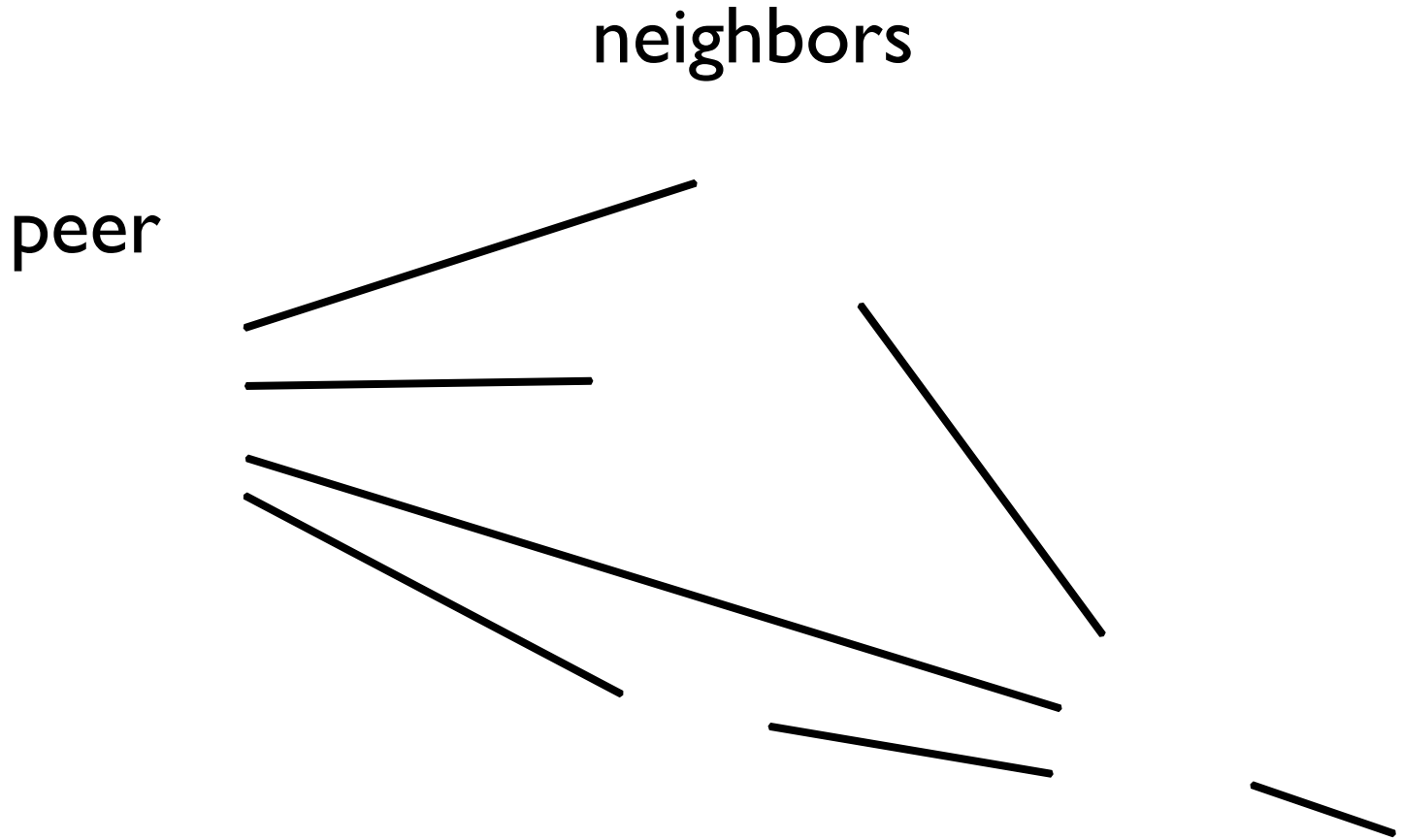
BitTorrent Overview



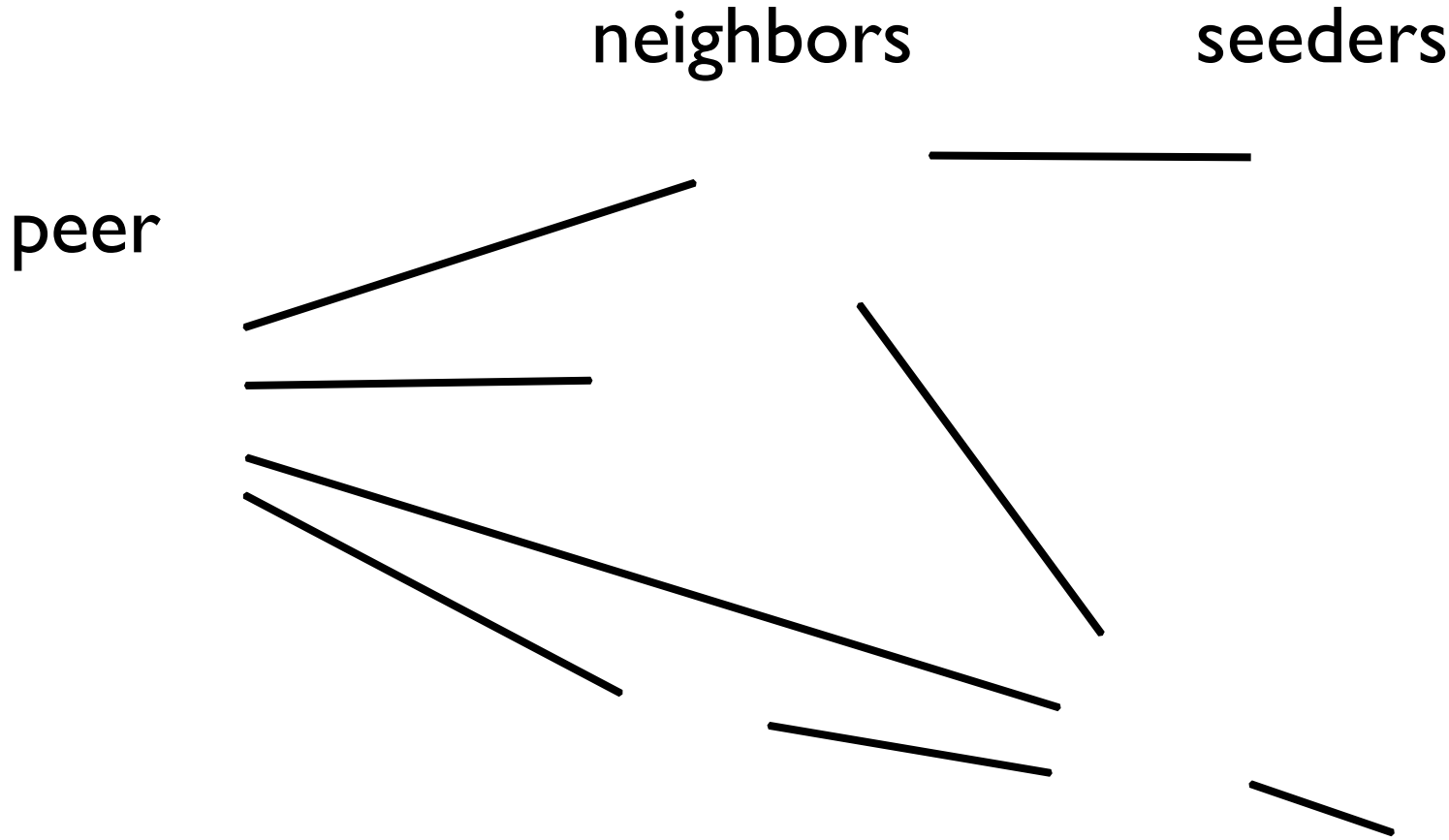
BitTorrent Overview



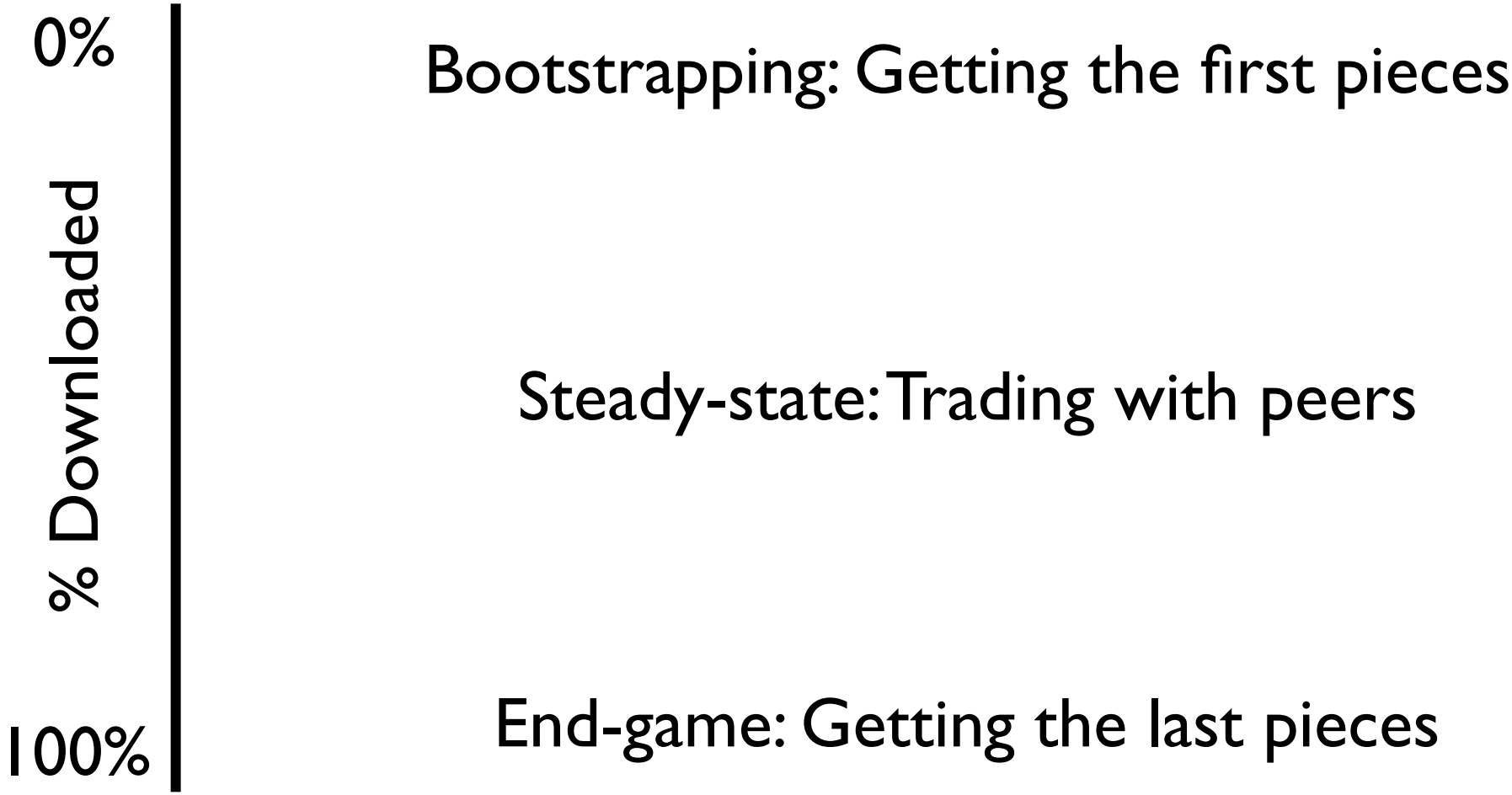
BitTorrent Overview



BitTorrent Overview



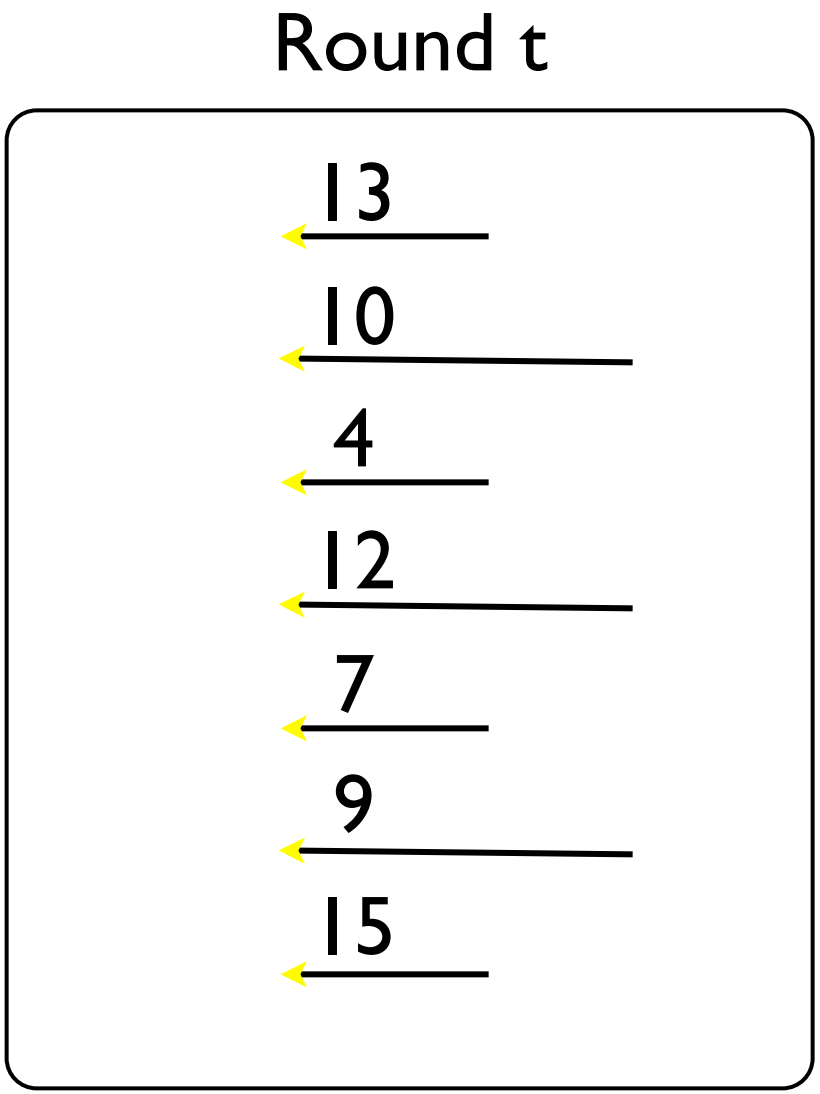
Phases of BitTorrent



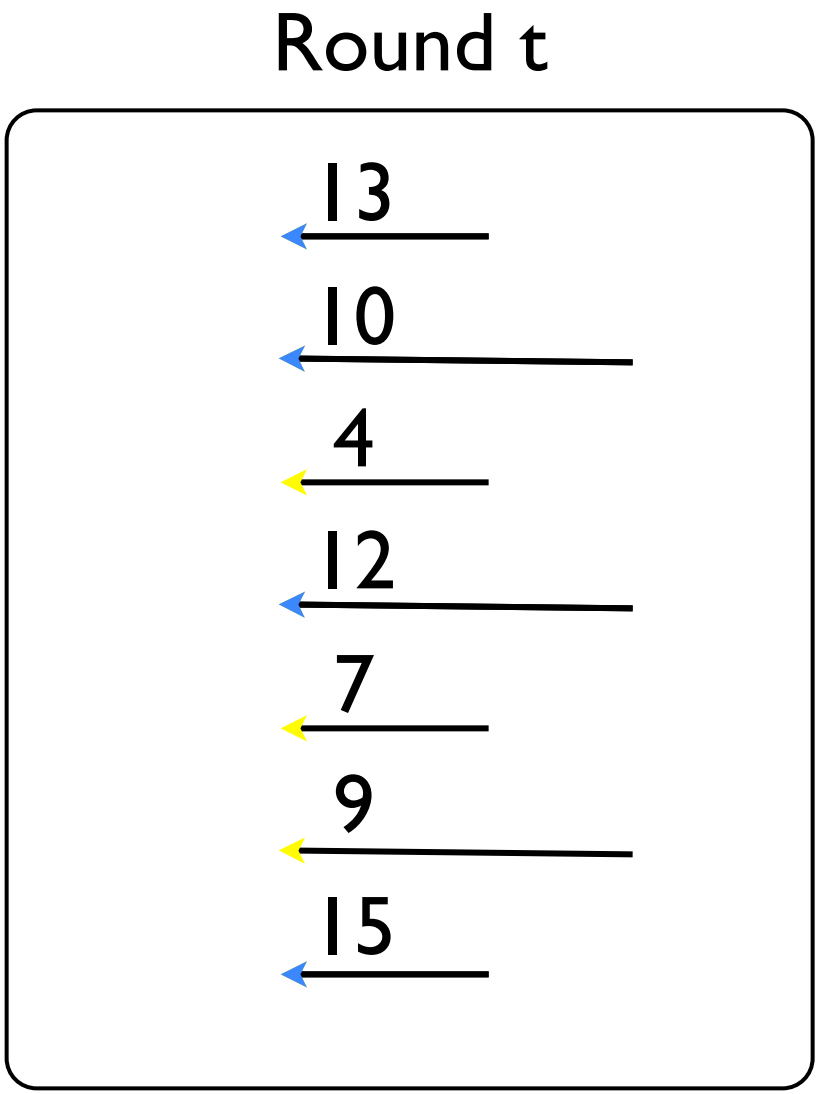
Incentive to Upload

- Every round, a BitTorrent peer calculates the number of pieces he has received from each of his neighbors
- The neighbors who gave the most are the ones who will get pieces in the next round
- The unchoker decides who to upload to, how much to upload, etc.

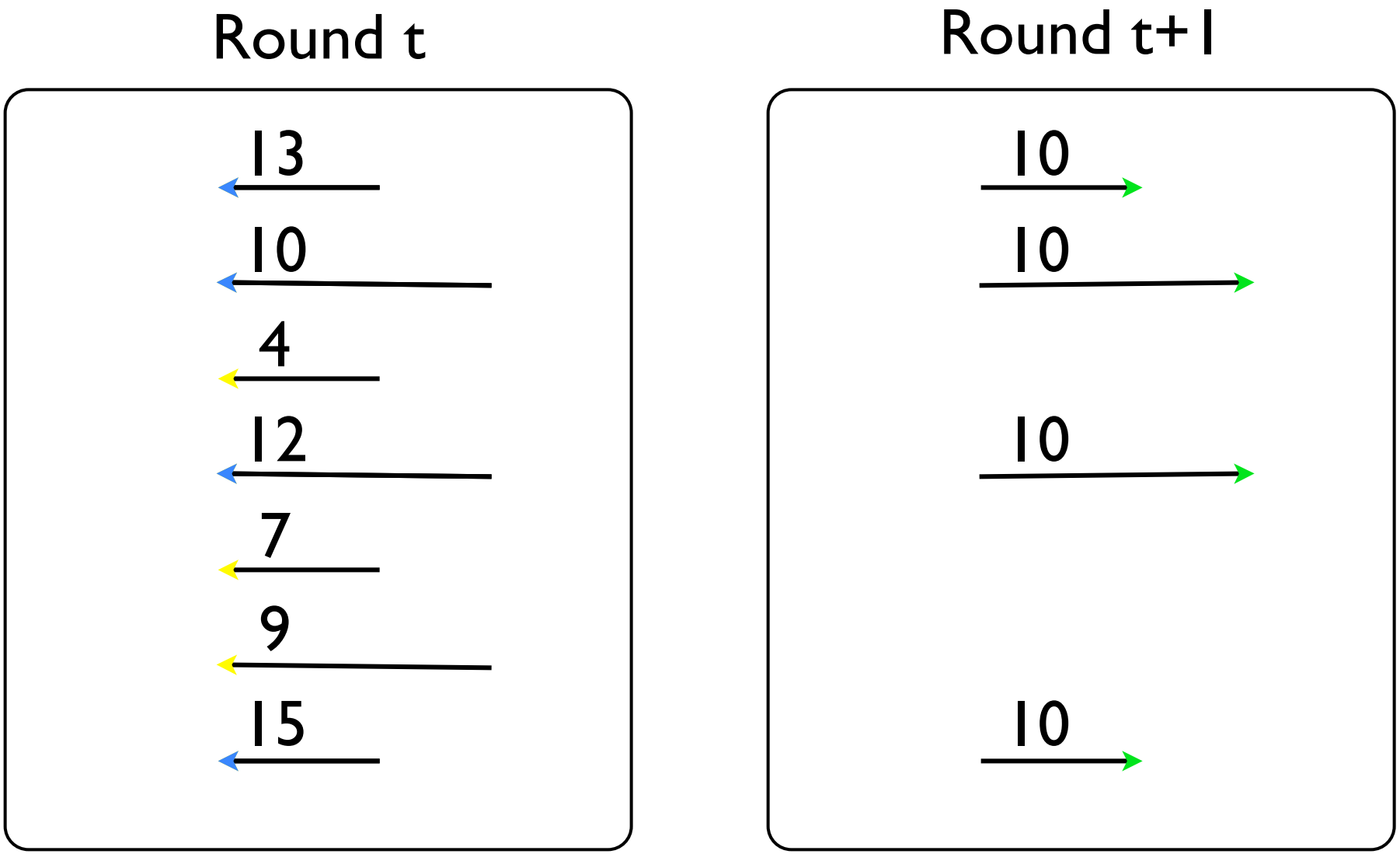
BitTorrent's Unchoker



BitTorrent's Unchoker



BitTorrent's Unchoker



BitTorrent's Unchoker

BitTorrent's Unchoker

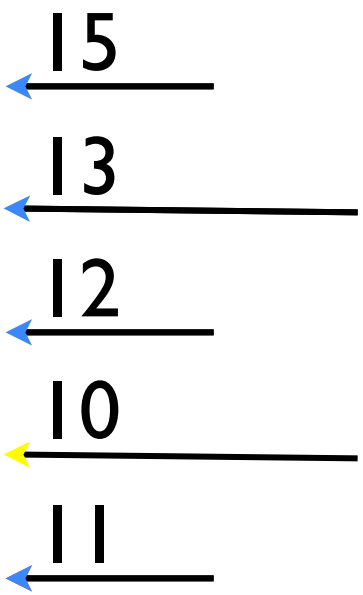
BitTorrent's Unchoker

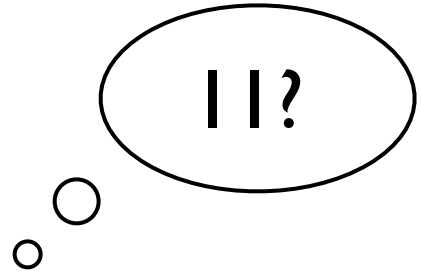
BitTorrent's Unchoker

BitTorrent's Unchoker

BitTorrent's Unchoker

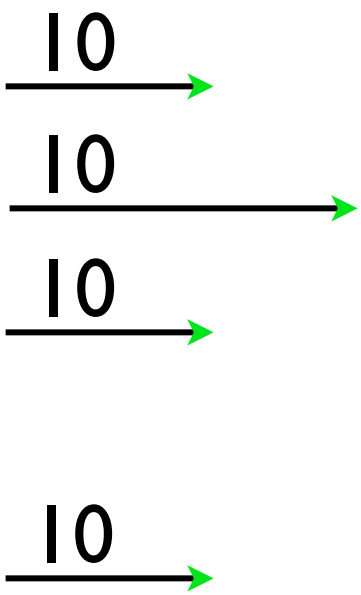
BitTorrent's Unchoker



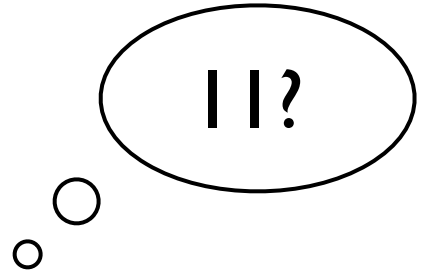


BitTorrent's Unchoker

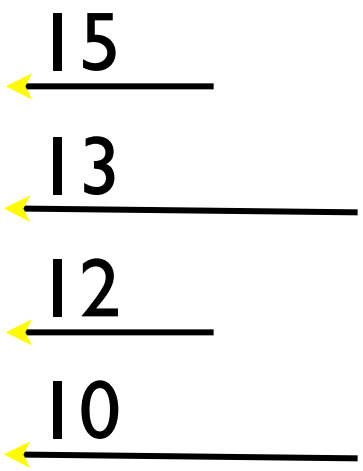
BitTorrent's Unchoker

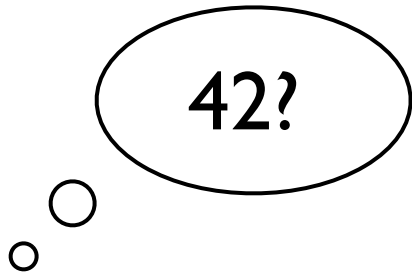


Best strategy: Come in fourth

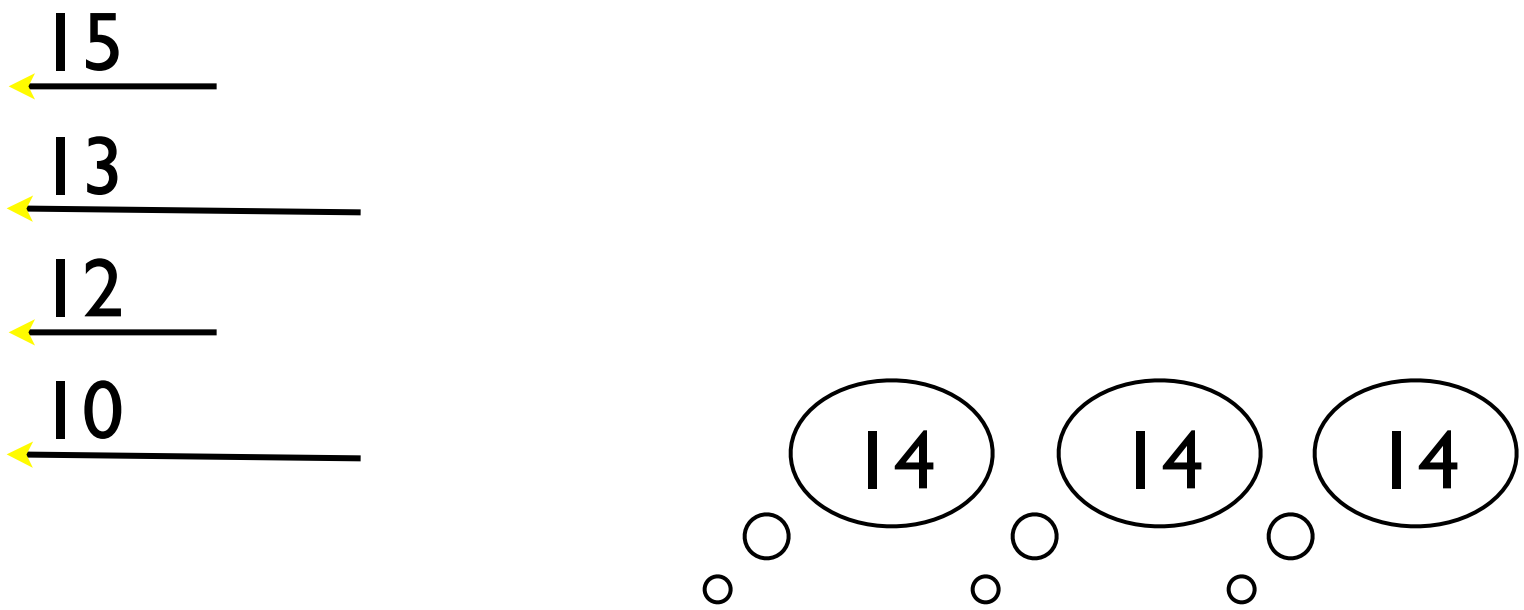


BitTorrent's Unchoker

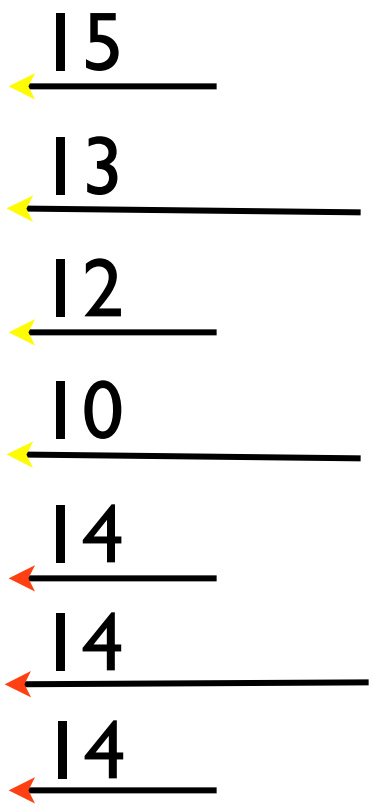


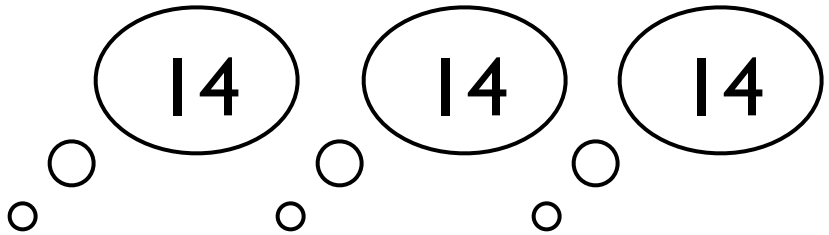


BitTorrent's Unchoker

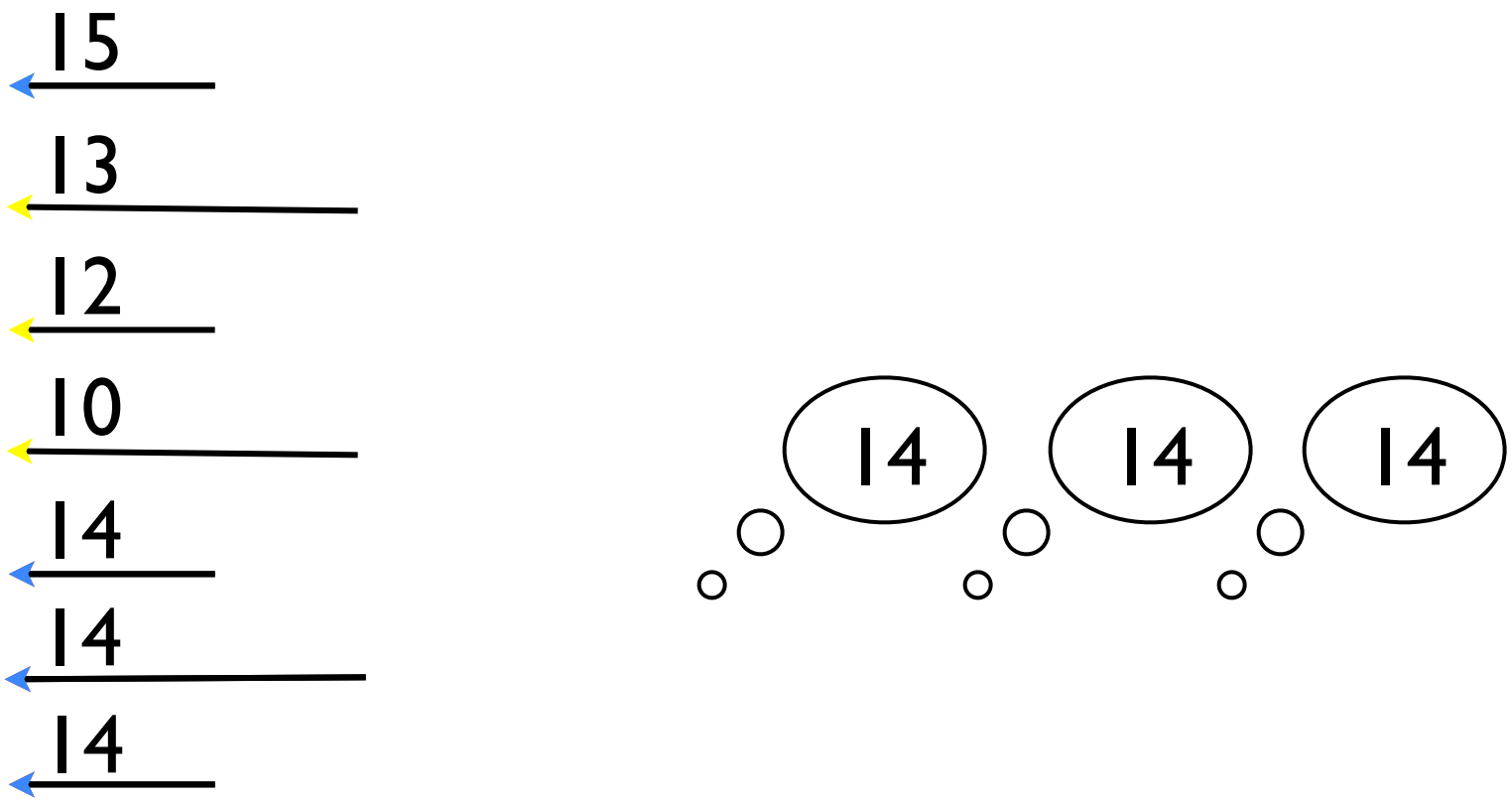


BitTorrent's Unchoker

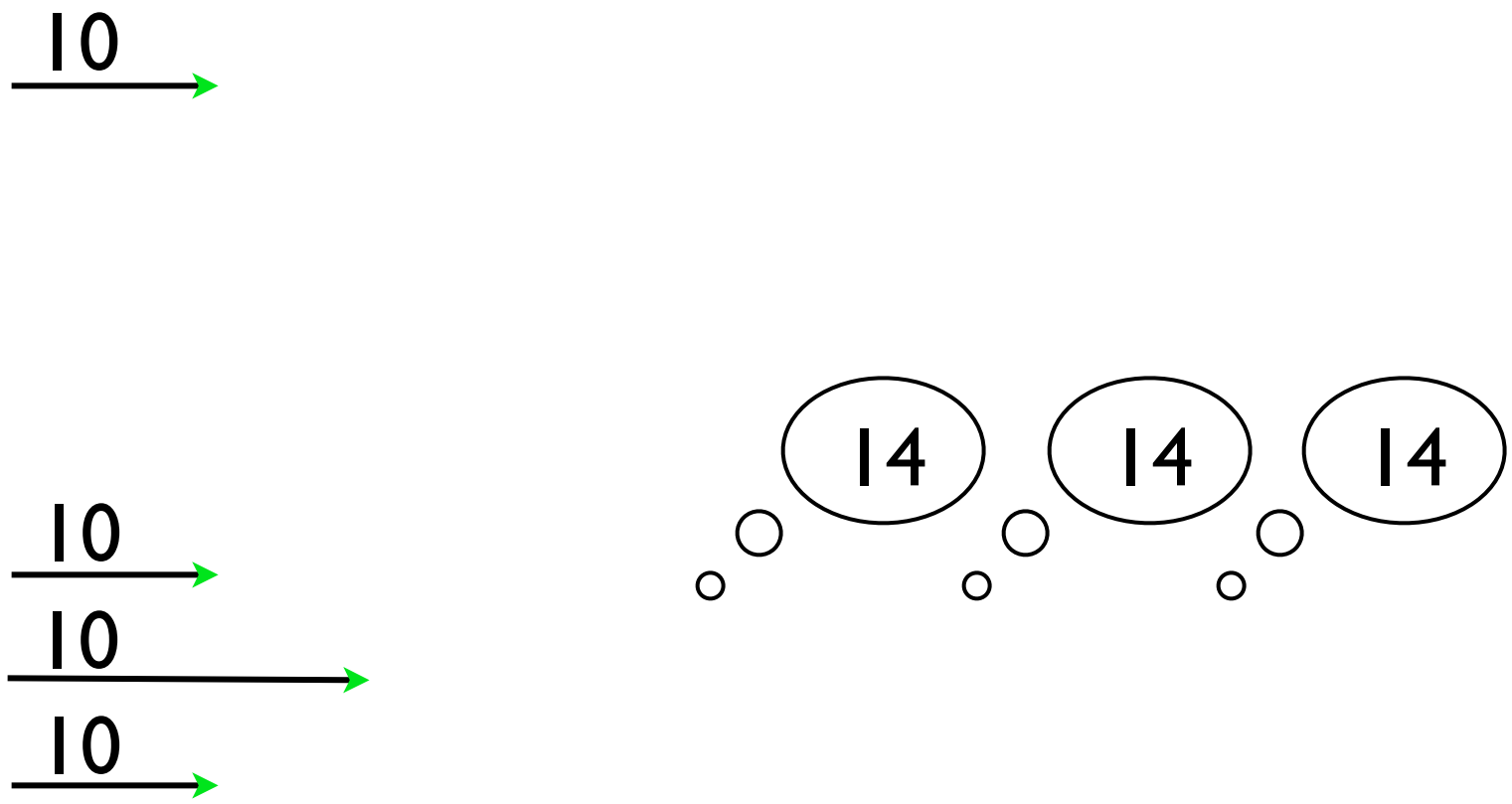




BitTorrent's Unchoker

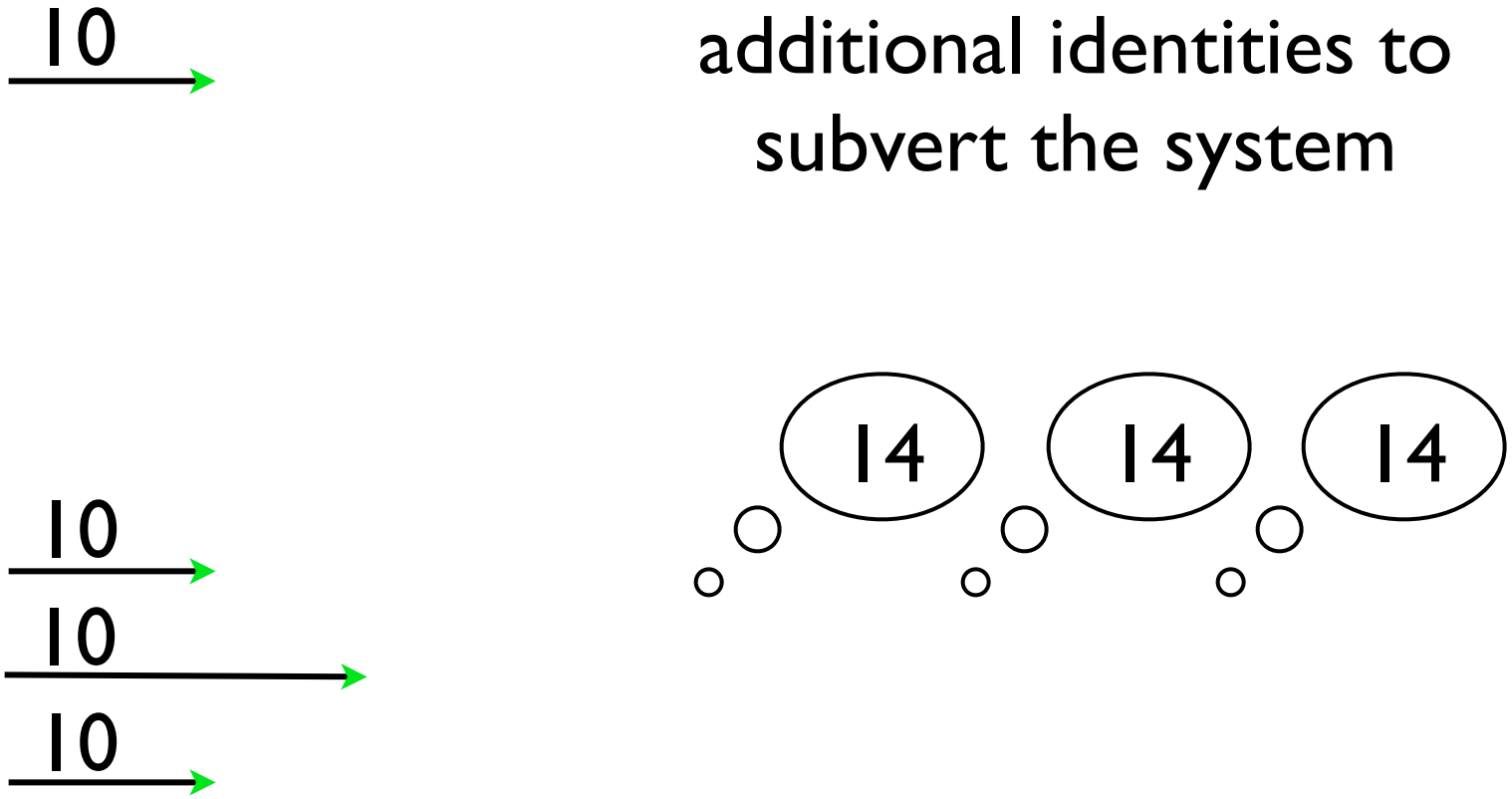


BitTorrent's Unchoker



BitTorrent's Unchoker

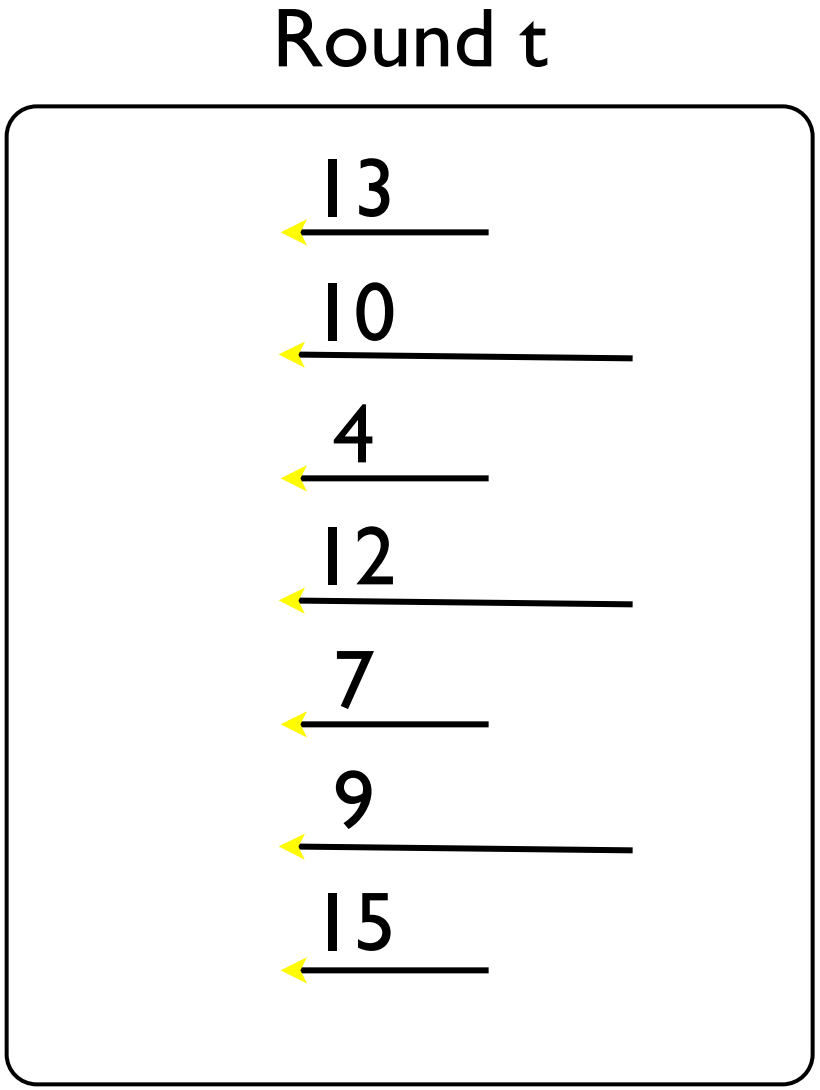
Sybil Attack: Create additional identities to subvert the system



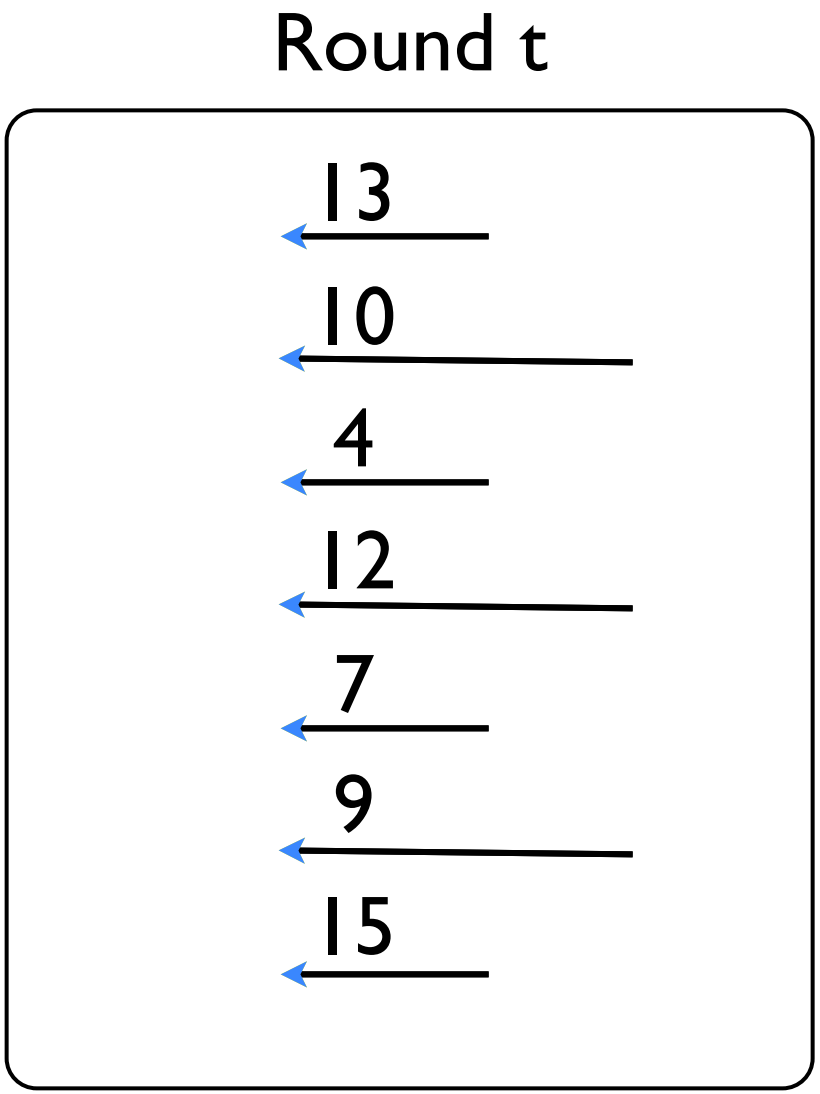
BitTyrant

- BitTyrant (Piatek et al., 2007) implements the "come in last" strategy
- Notice that peers *don't* have incentive to give as much as possible
- Can we come up with a strategy that is Sybil-proof, fair, and encourages peers to give as much as they can?

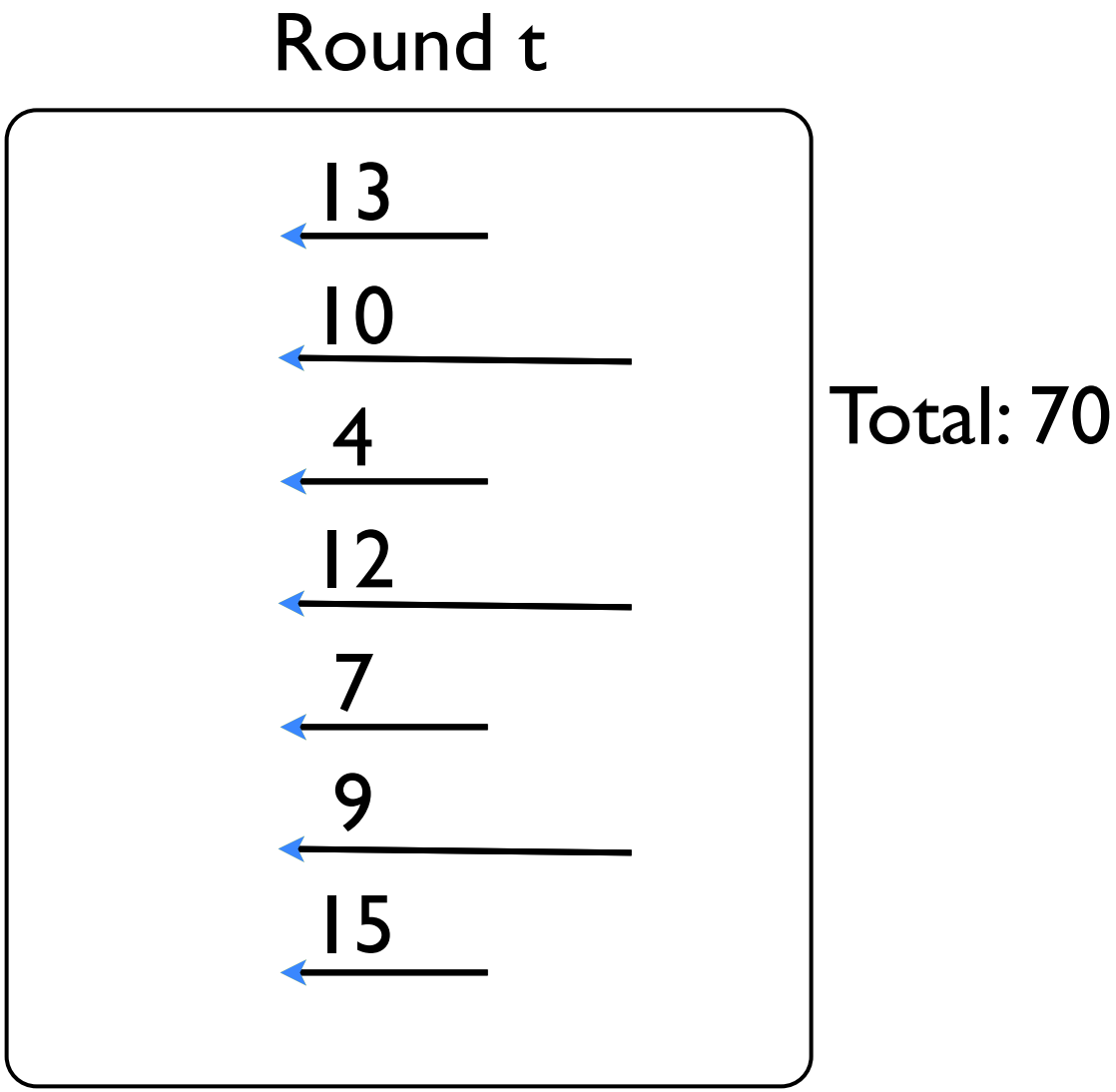
PropShare Unchoker



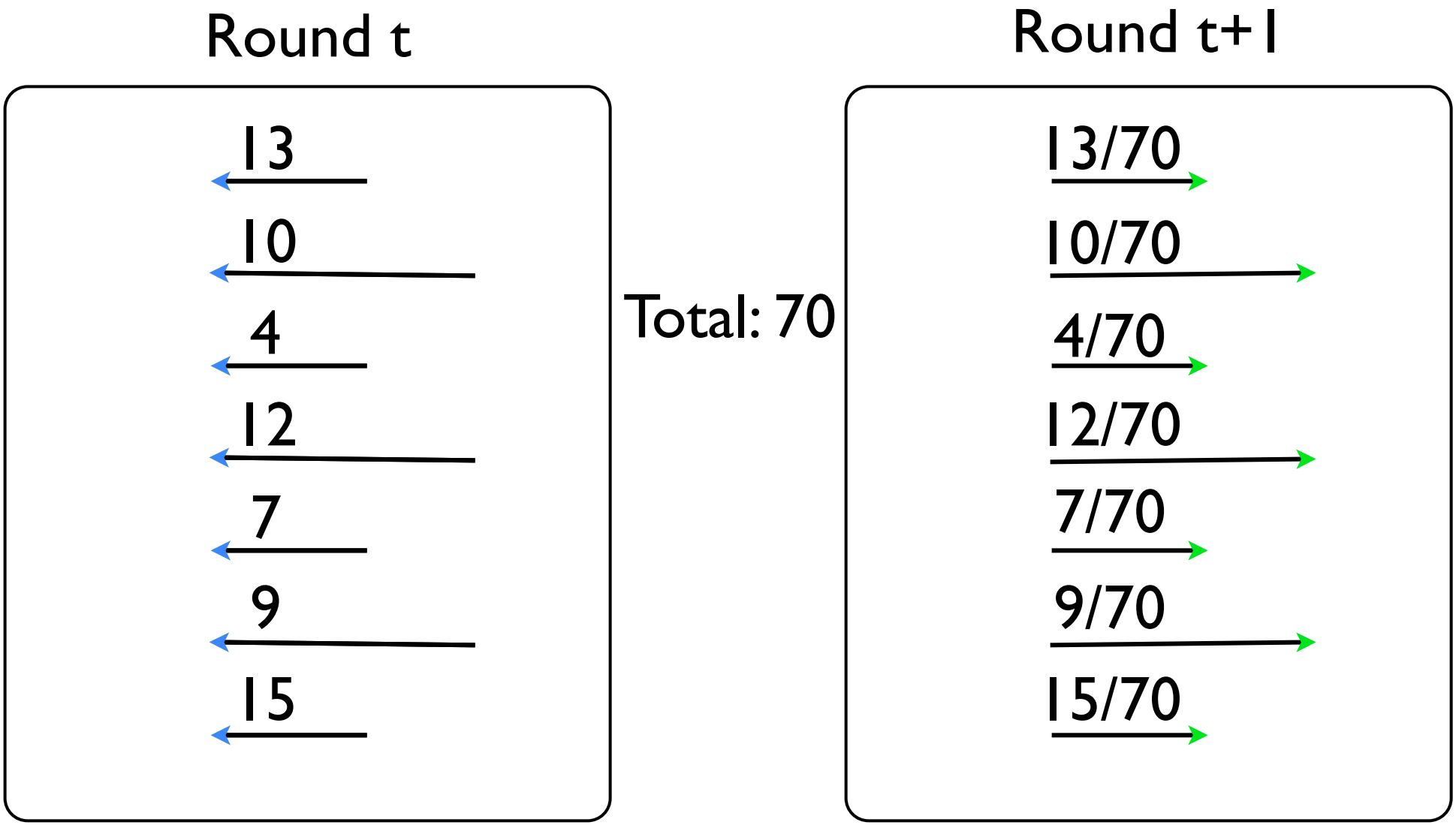
PropShare Unchoker



PropShare Unchoker



PropShare Unchoker



PropShare Unchoker

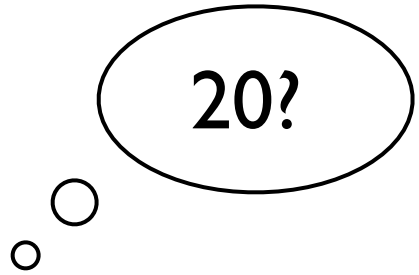
- A PropShare client responds by allocating its upload bandwidth *proportionally* to its neighbors
- Consequentially, there is no longer a "top four"
- Can we cheat in the same way we did before?

PropShare Unchoker

PropShare Unchoker

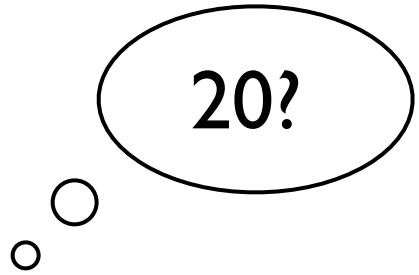
PropShare Unchoker

Total: 70



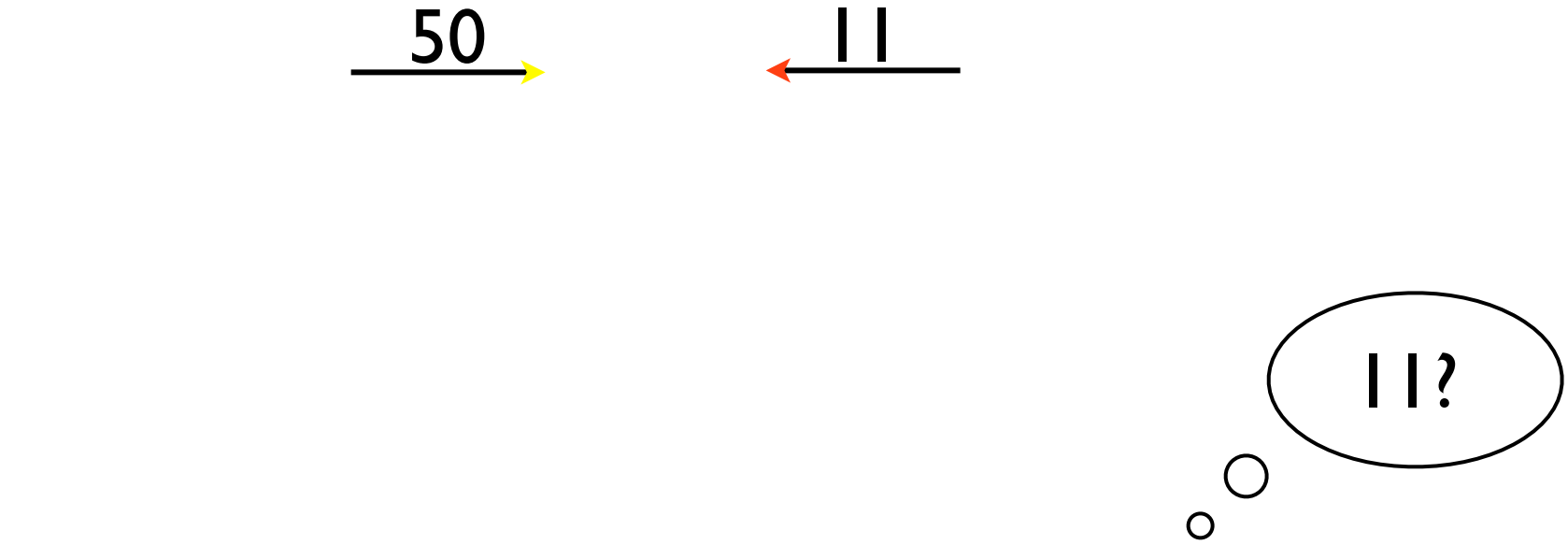
PropShare Unchoker

Total: 70



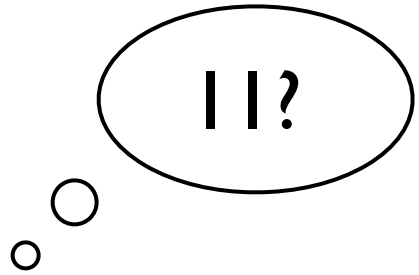
PropShare Unchoker

PropShare Unchoker

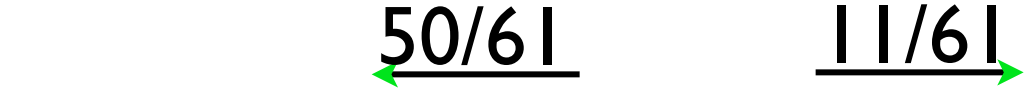


PropShare Unchoker

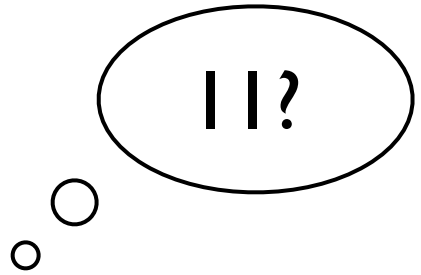
Total: 61



PropShare Unchoker



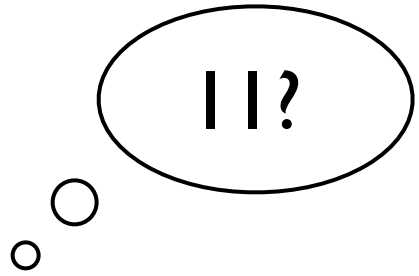
Total: 61



PropShare Unchoker

50/61 ← → 11/61

Total: 61

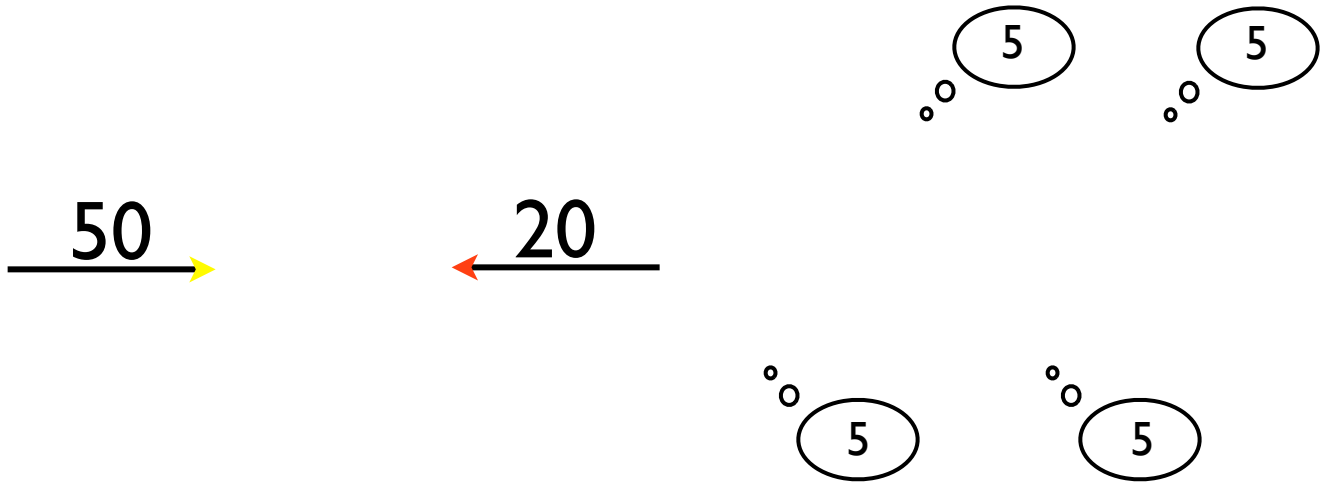


Upload Less →

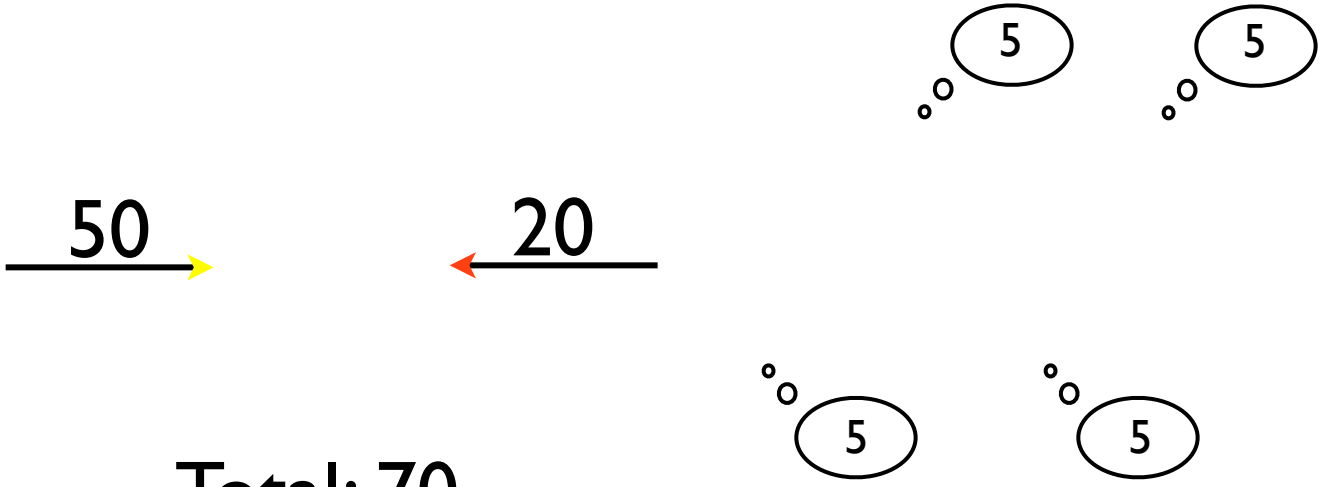
Receive Less →

Incentive to Upload More

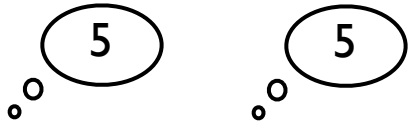
PropShare Unchoker



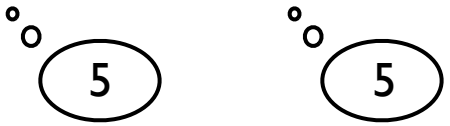
PropShare Unchoker



PropShare Unchoker

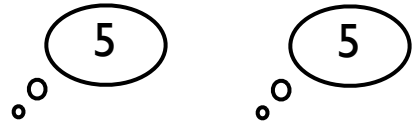


50/70 20/70



Total: 70

PropShare Unchoker



50/70 20/70

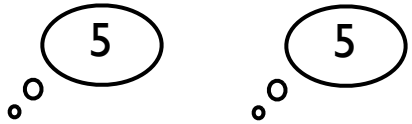
Total: 70

20

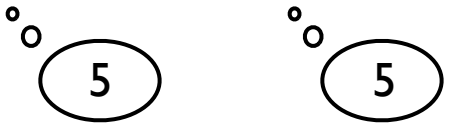
5 5 5 5

\+ + + =

PropShare Unchoker



50/70 20/70



Total: 70

20

5 5 5 5

\+ + + =

PropShare is Sybil-proof

Unchoker Results

- BitTyrant and PropShare are both faster than BitTorrent
  - For different reasons
- PropShare performs comparably to BitTyrant
- PropShare does *not* suffer from a tragedy of the commons; BitTyrant does

Steady-state Summary

- The BitTorrent unchoker can significantly impact download times
- Unfair unchokers (e.g., BitTyrant) tend to suffer as more peers use them
  - But in a swarm of mostly non-BitTyrant peers, BitTyrant performs very well

Bootstrapping

- We've been assuming that all peers have some part of the file to trade (steady-state)
  - They can trade blocks with other peers because they are *interesting*; they have blocks to offer
- What about when a peer has nothing to trade?
  - The bootstrapping mechanism determines what to do in this situation

Optimistic Unchoking

- Reserve a certain portion of our bandwidth to give *freely* to other peers (presumably new peers)
  - This can clearly be exploited
- BitThief (Locher et al., 2006) peers upload *nothing*
  - System will collapse if all peers use BitThief

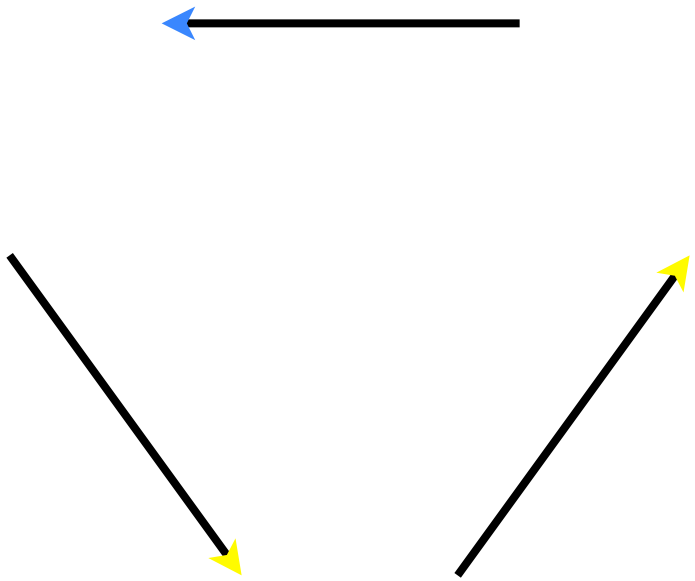
Other Bootstrapping Mechanisms

- Junk updates: peers upload fake data, proving that they're willing to do work
  - Can't be exploited in the same way as optimistic unchoking, but wastes system resources
- TBS: put new peers to work by forwarding data for other peers

TBS

TBS

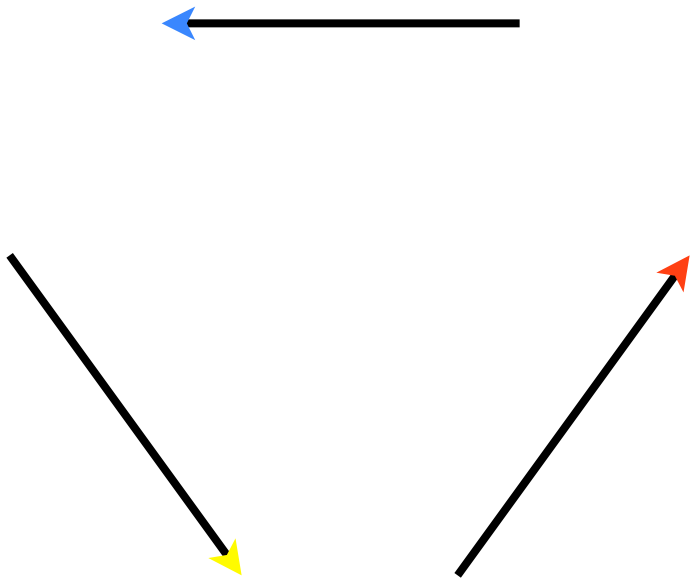
TBS



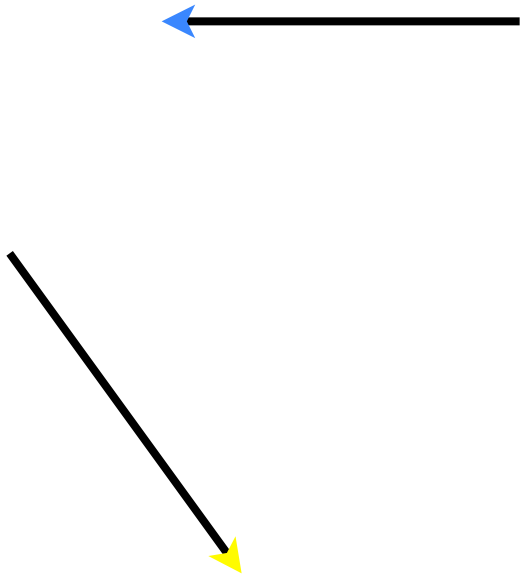
TBS

TBS

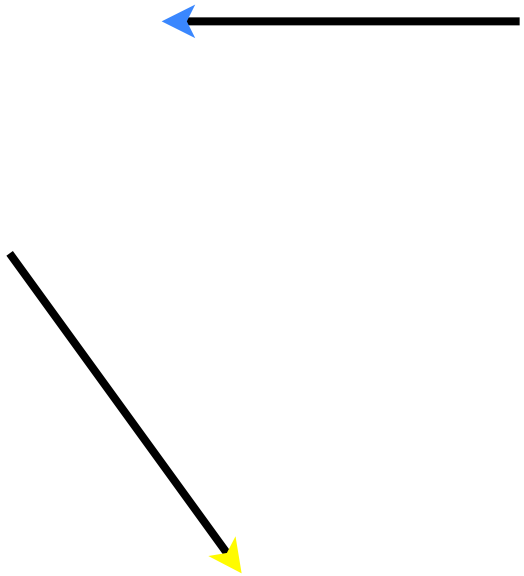
TBS



TBS

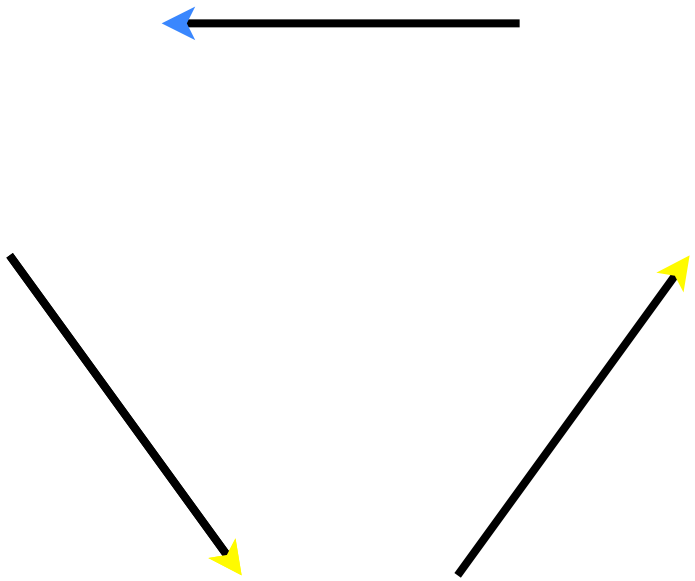


TBS

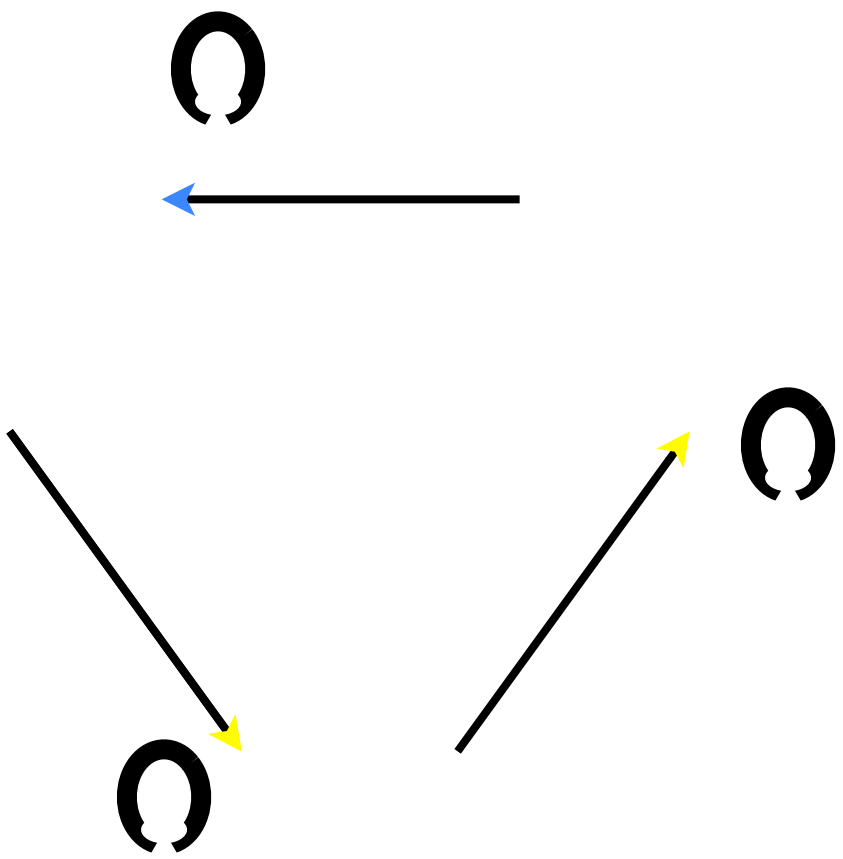


Solution: Encryption

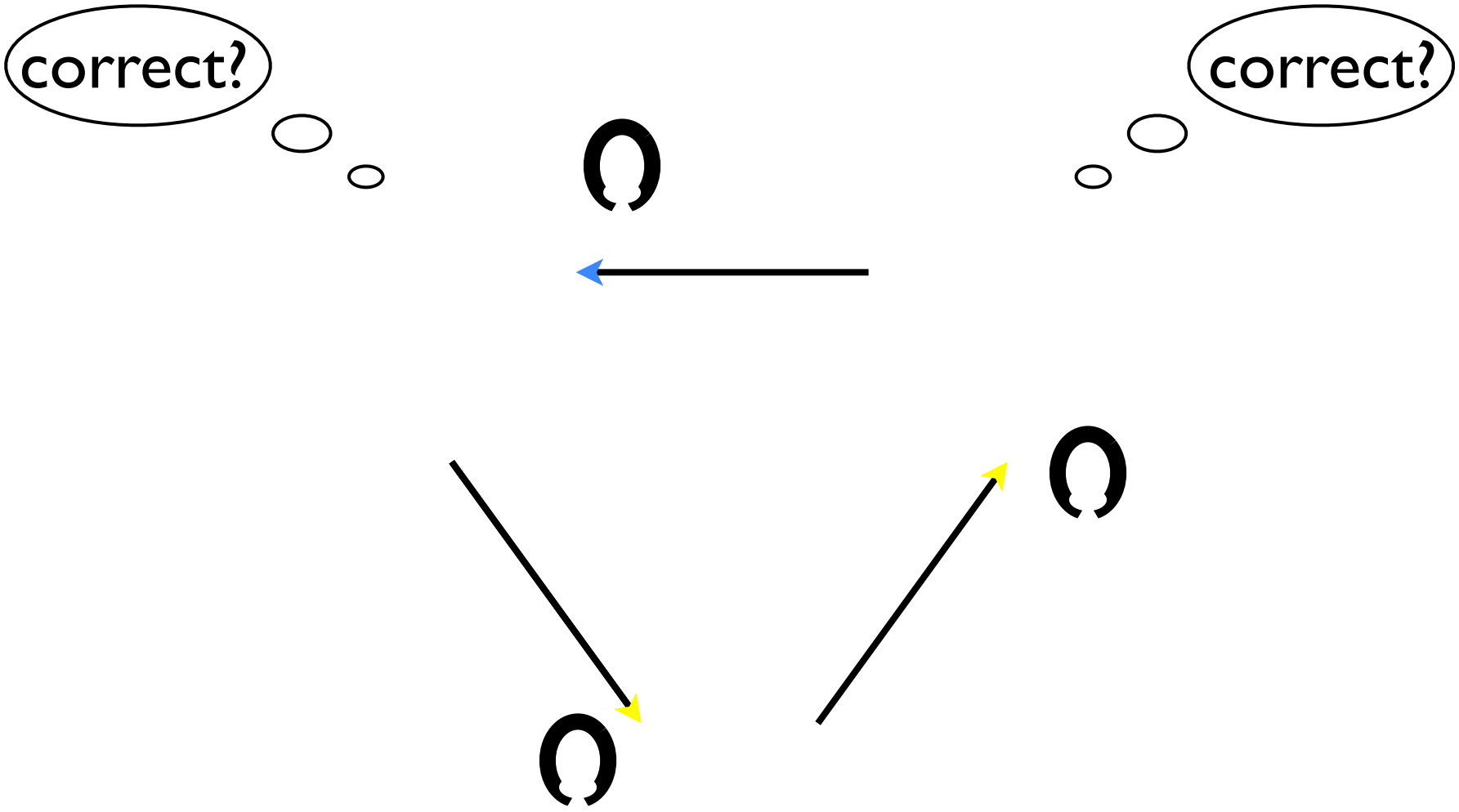
TBS



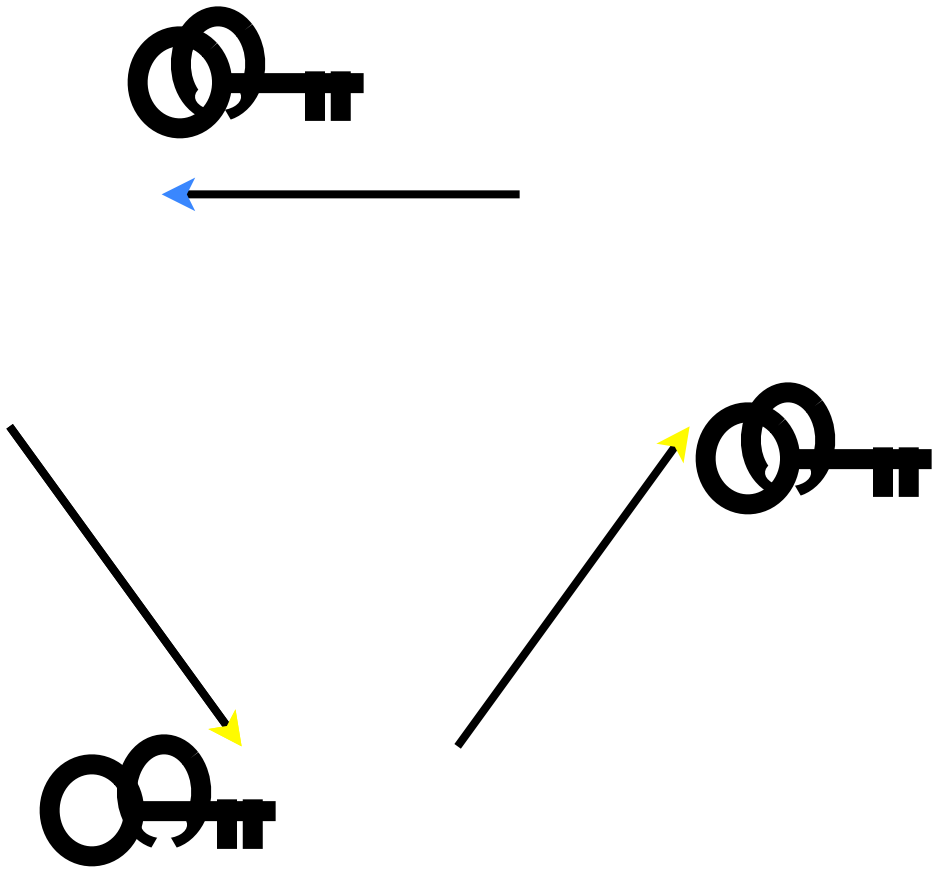
TBS



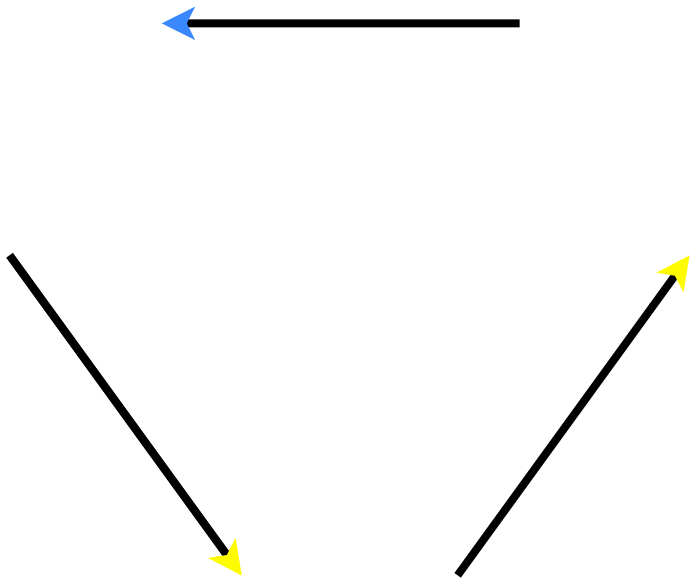
TBS



TBS



TBS



Bootstrapping Summary

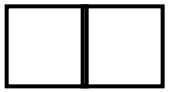
- Bootstrapping is not a very large part of the download. Even so, it can be exploited
- A better bootstrapping mechanism has potential to yield better performance *throughout* the download
  - Moreover, it can be used whenever a peer becomes uninteresting, not just in the bootstrapping phase

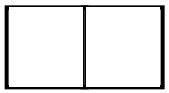
End-game

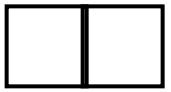
- During end-game, we only need a few pieces, and so are not interested in many peers
  - Might end up just going to seeders
- Peers want to *maintain interest*, i.e., remain interesting to as many peers as possible
- Interest is determined by examining a peer's bitmap, which indicates which pieces of the file they have

Strategic Piece Revelation

Round t

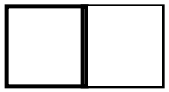


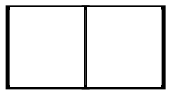


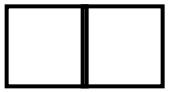


Strategic Piece Revelation

Round t

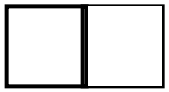


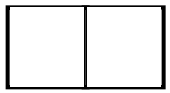


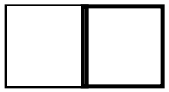


Strategic Piece Revelation

Round t

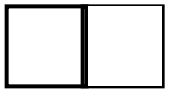


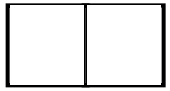


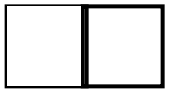


Strategic Piece Revelation

Round t+1

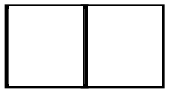


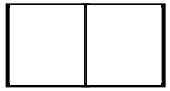


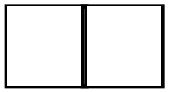


Strategic Piece Revelation

Round t+1

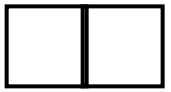


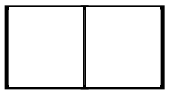


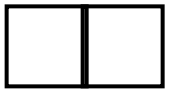


Strategic Piece Revelation

Round t

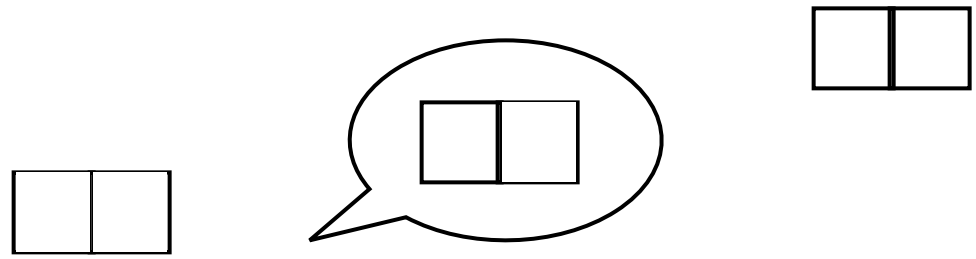






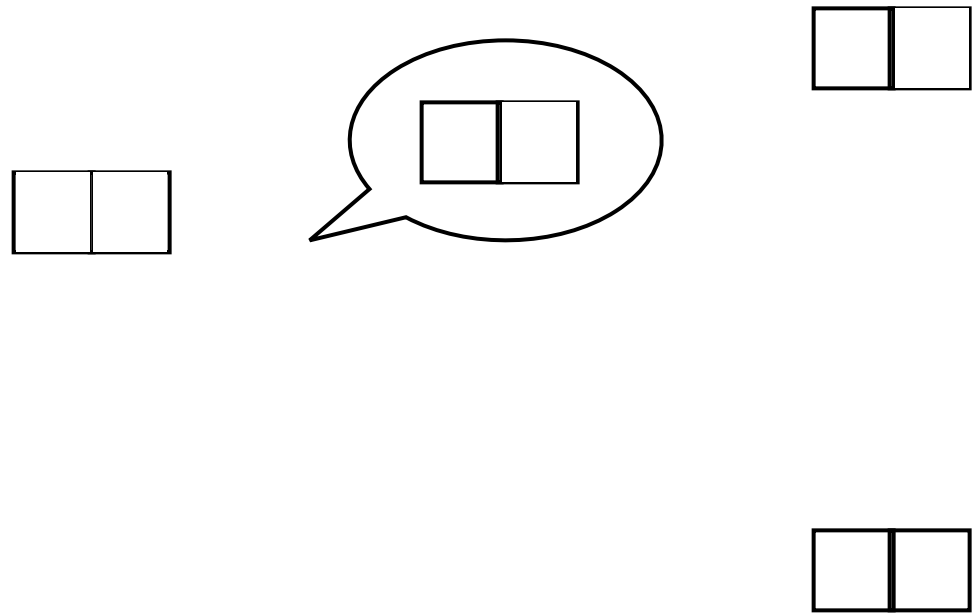
Strategic Piece Revelation

Round t



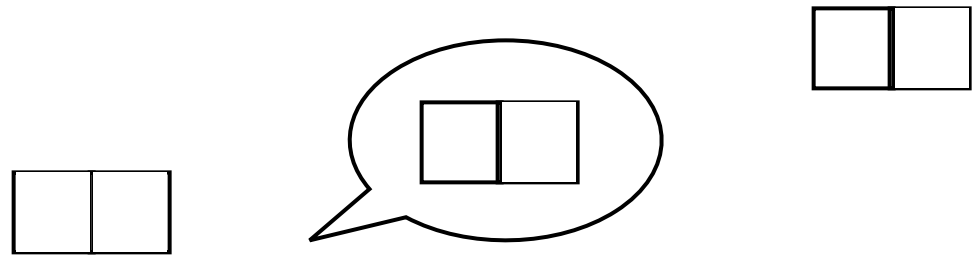
Strategic Piece Revelation

Round t



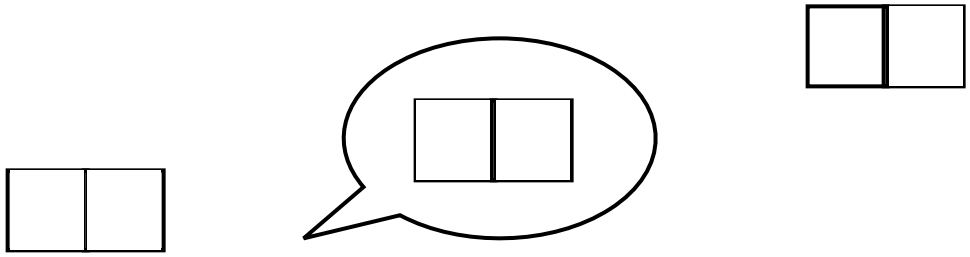
Strategic Piece Revelation

Round t



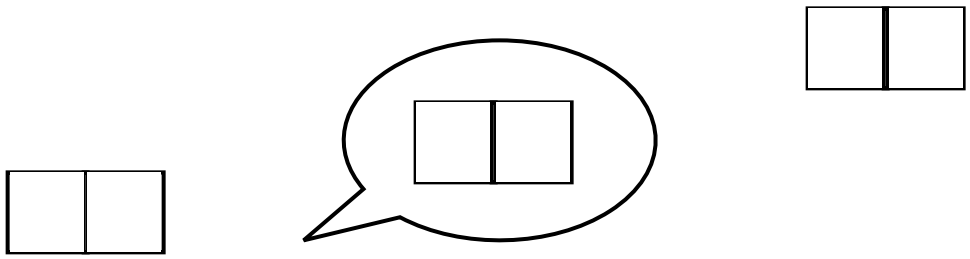
Strategic Piece Revelation

Round t+1



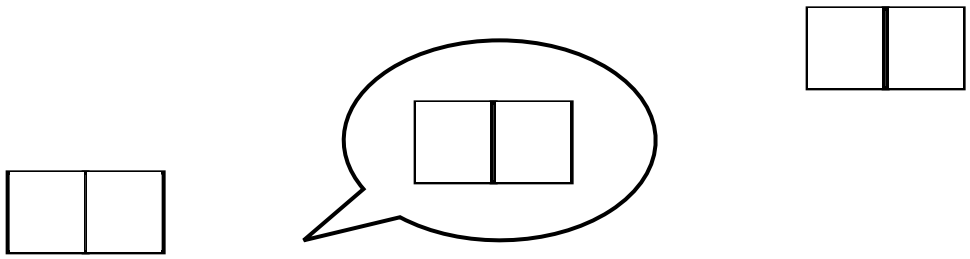
Strategic Piece Revelation

Round t+1



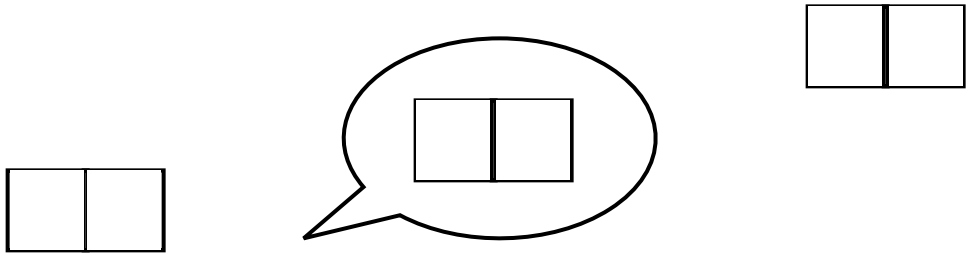
Strategic Piece Revelation

Round t+1



Strategic Piece Revelation

Round t+1



Strategically reveal pieces →
Peers are interested in me longer

Piece Revelation Results

- When one peer strategically reveals, he remains interesting longer, and downloads faster
- When *all* peers strategically reveal, total download time increases

Peer Selection

- Before our download even starts, a BitTorrent client gets a set of peers from the tracker.
- During the download, the peer figures out the "best" of this set
- What if we could decide which peers would be best without trading with them first?

Peer Selection

- Measuring link characteristics is sometimes seen as a threat, and doesn't scale
- Many measurement systems require a "map" of the Internet, which is hard to obtain
- Network coordinate systems don't require a map, but are complicated and don't always work
- Could try simple things (use peers in our ISP, e.g.), but it's not clear that these work either

Summary

- BitTorrent is a large systems; lots of things to tweak
  - Bootstrapping, steady-state, end-game phases
  - Peer selection
- Not all strategies are fair
- A combination of techniques (from various phases) would probably result in an extremely fast client

Thanks

- Slides adopted from Dave Levin's SIGCOMM 2008 talk
- Work on PropShare and TBS done with Dave Levin, Neil Spring, and Bobby Bhattacharjee at the University of Maryland