Leveraging Altruistic Peers to Reduce the Bandwidth Costs of Free Content Downloads

Cameron Dale (camerond@cs.sfu.ca) and Jiangchuan Liu, Simon Fraser University

Many Current Systems

- files are available to everyone for free
- cryptographic hash of the file is available before downloading
- all content is divided into packages
- altruistic users exist who are willing to contribute upload bandwidth but have no way to do so

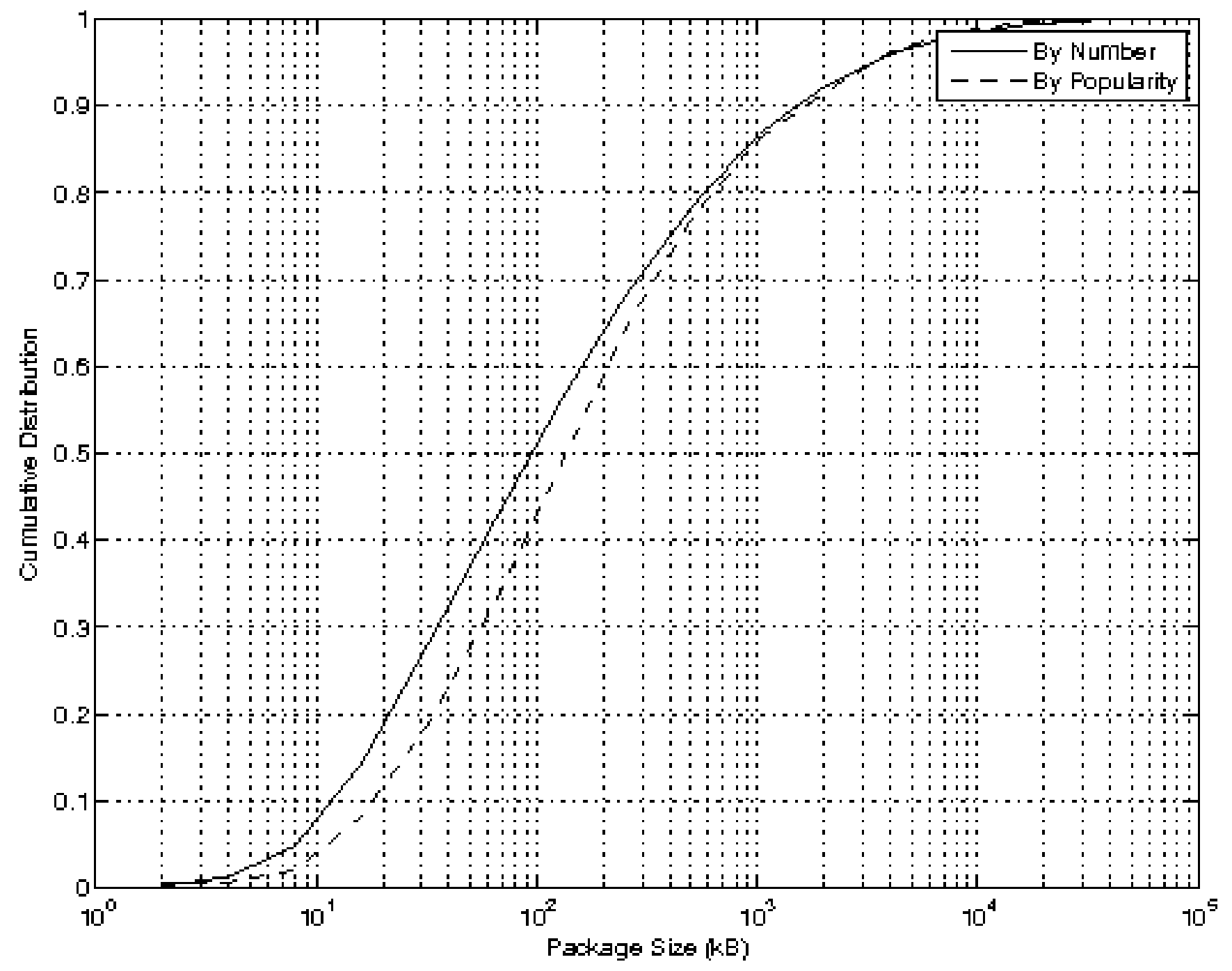
A few of the groups that could benefit

Problems to be Solved

- users are only interested in a small percentage of the total packages available

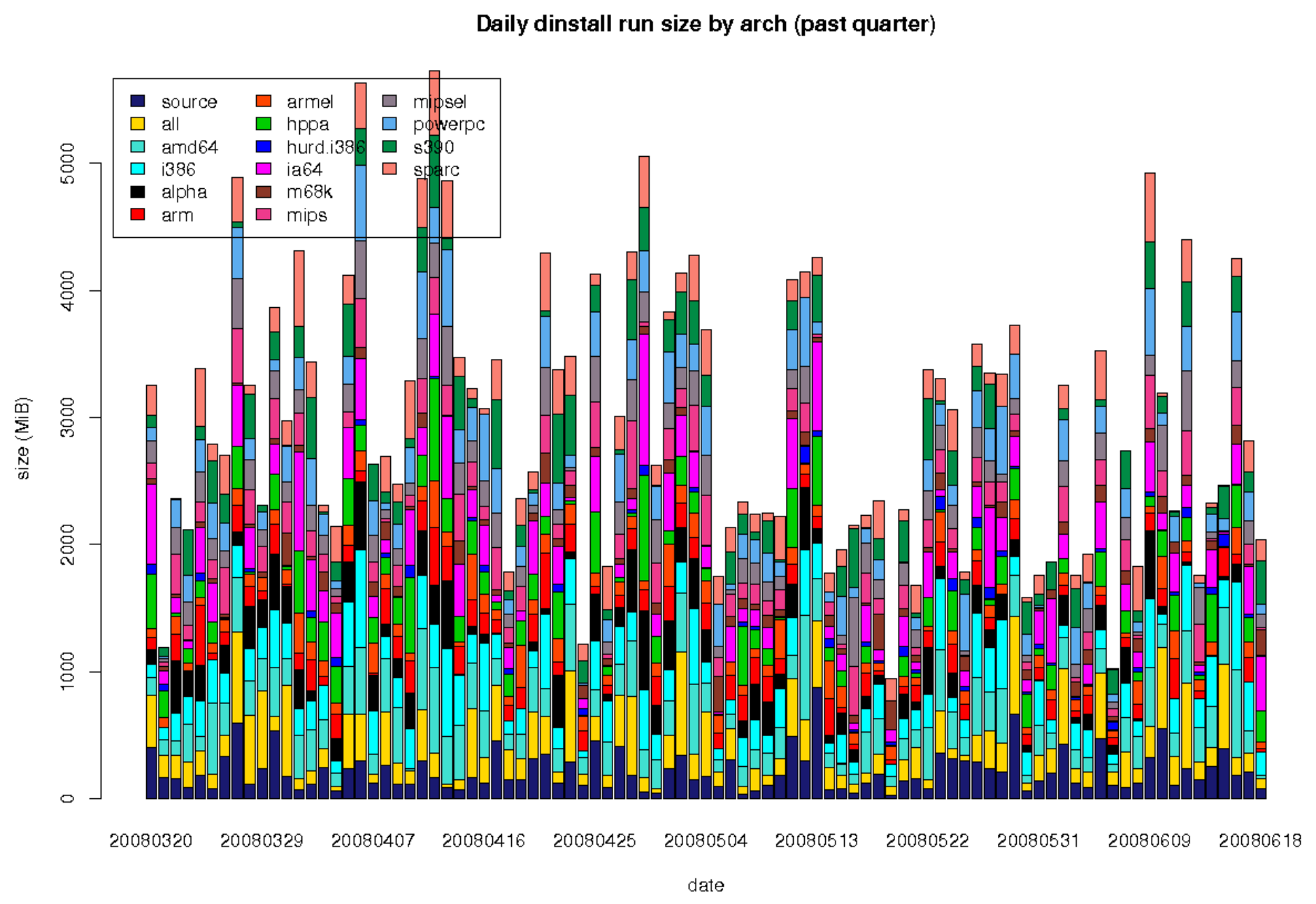
Debian users are typically only interested in less than 5% of the 23,000 packages available to them.

- packages are mostly small in size, but some can be very large



CDF of Debian packages' size, both by the number of packages, and taking into account the popularity of each package. 80% of packages are less than 1 MB, but a few packages are hundreds of MB.

- a small percentage of packages are updated regularly (daily)

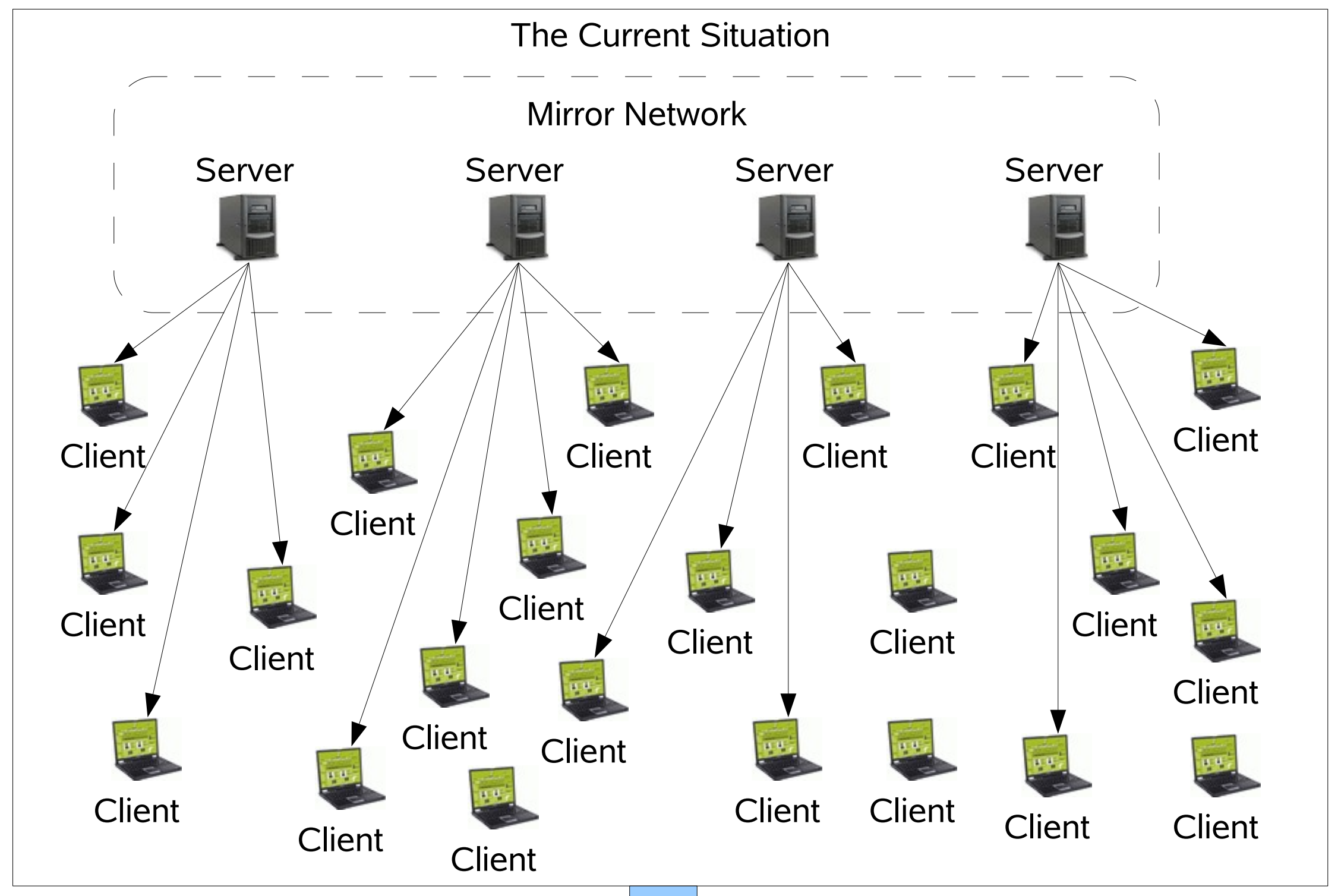


The size of Debian's daily archive updates, broken up by architecture. Approximately 1% (by size) of the 119,000 MB archive is updated every day.

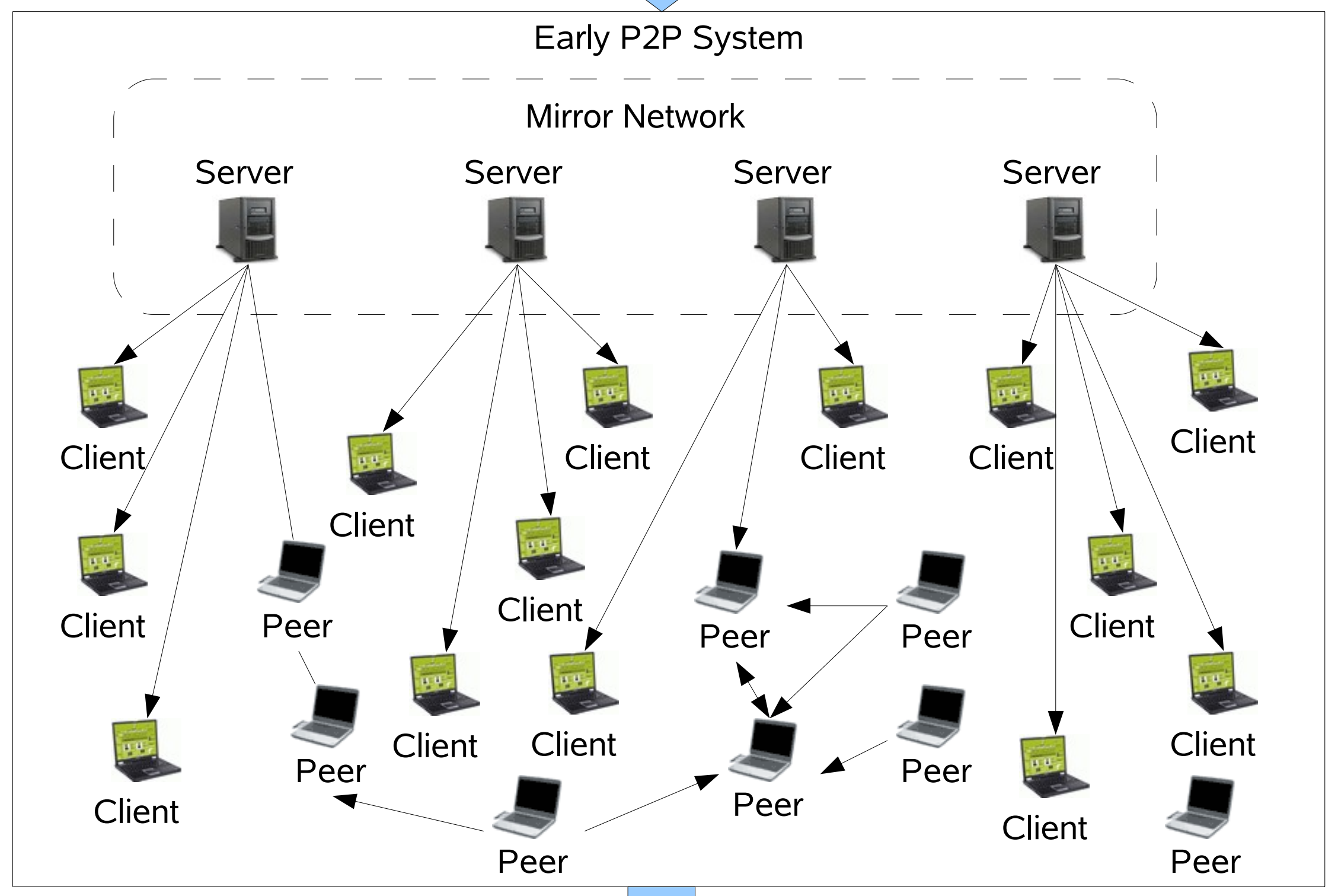
Requirements

- simple to implement, as unique implementations are required for different systems
- work with the existing mirrors unmodified
- deploy incrementally with no user dissatisfaction
- no undue burden is placed on any peers
- fast lookup times to support interactive downloading
- fast download times using techniques borrowed from other P2P programs (BitTorrent)

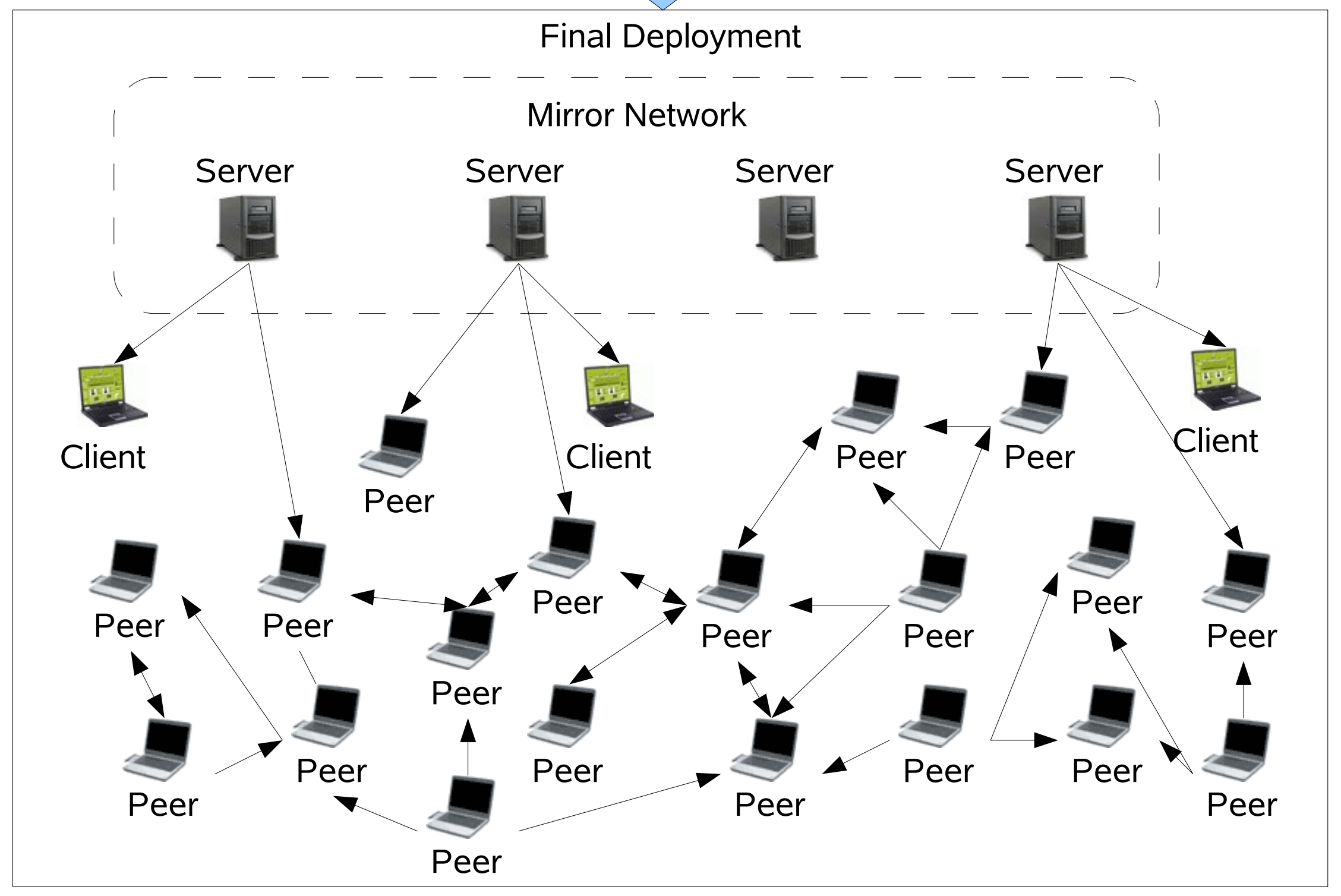
Incremental Deployment



Incremental deployment as users slowly adopt the new P2P system.
During the deployment the users are still fully functional.

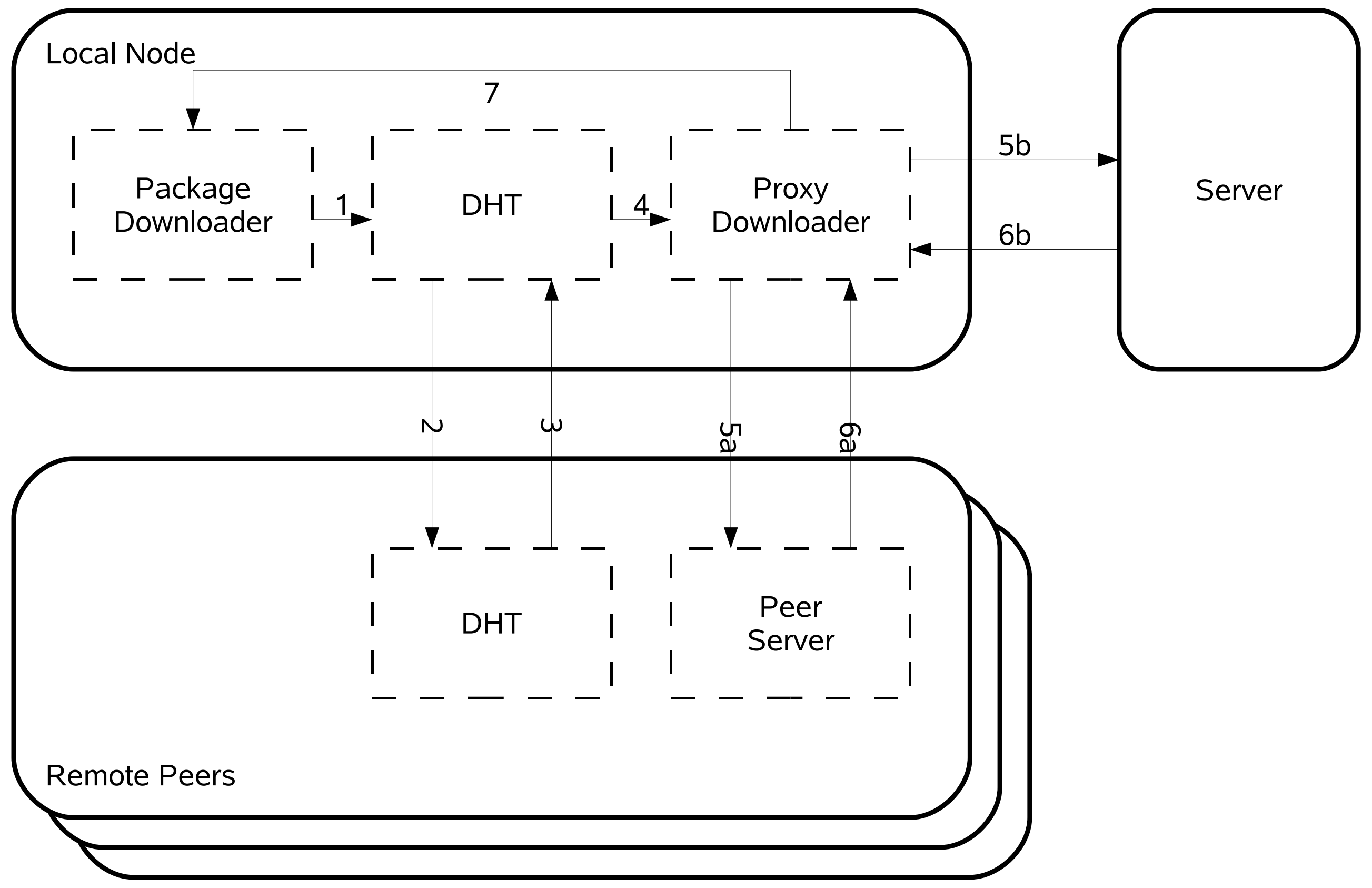


Eventually word spreads and the P2P system gains in popularity.
Server bandwidth is reduced, but some servers will always be required.



Proposed Model

- integrates with the traditional package downloader by operating as a proxy between the package downloader and the server/peers (1, 7)
- peers that are sharing individual packages are found using lookups of the cryptographic hash of the package in the DHT (2, 3)
- packages are downloaded from peers in parallel, using multiple peers and breaking up larger packages into pieces for efficiency (5a, 6a)
- packages that can not be found in peers fall back to downloading from the server, so the server operates as a seed in the system for new or rare packages (5b, 6b)
- peer information (and piece information) is stored in the DHT indicating that this peer now has the package available to share



Customized DHT

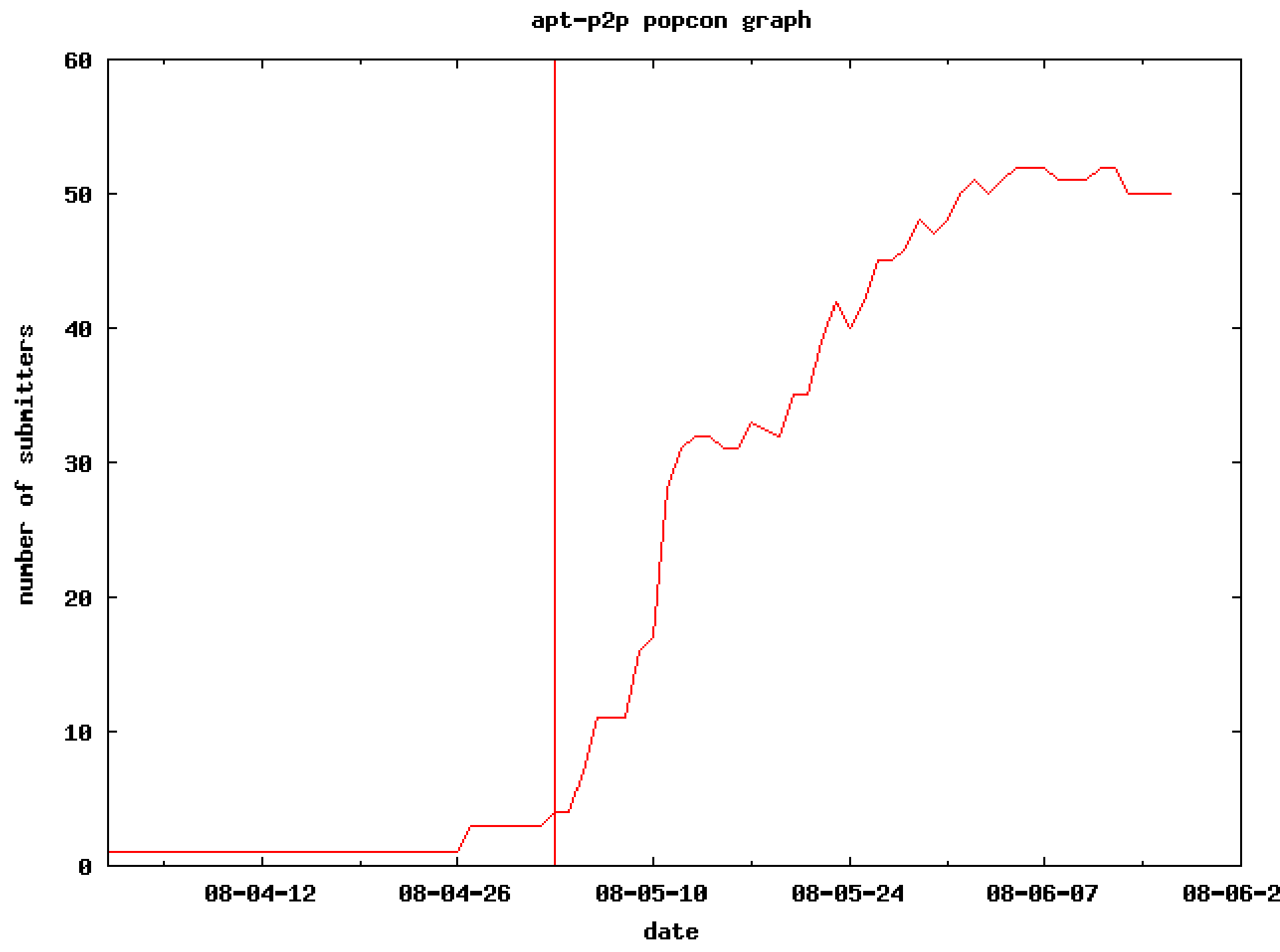
- modified version of Kademlia using ideas from the BitTorrent tracker-less DHT
- everything is stored as bencoded dictionaries (like BitTorrent), making enhancements easy
- peers store their location (IP/port) value at the key that is the cryptographic hash of the package
- modified to support multiple values (peers) per key (package)
- improved lookup times to support interactive downloads (still needs work)
- large packages are broken up into 512 kB pieces, and the piece hashes are stored in the DHT

Piece Hash Storage Strategy

- a single piece
  - no piece hashes are needed
  - peers store only their location
- a few pieces
  - hash the pieces of the package
  - store the piece hashes with the peer location at the package's hash key
- 10's of pieces
  - too many pieces to store with the peer location
  - hash the list of piece hashes to get a piece hash key
  - store the list of piece hashes at the piece hash key
  - store the piece hash key with the peer location at the package hash key
- 100's or 1000's of pieces
  - too many pieces to store in the DHT
  - hash the list of piece hashes to get a piece hash key
  - save the list of piece hashes so others can request it using the piece hash key
  - store the piece hash key with the peer location at the package hash key

Example Program: apt-p2p

- caching HTTP proxy for Debian's APT package download program
- DHT is based on Khashmir
- all peers are HTTP/1.1 servers, which support pipelining multiple requests and Range requests for pieces of a package
- servers are also HTTP-based, and so are used almost identically as peers
- implementation is available for any Debian user to install in the apt-p2p package



The popularity of the apt-p2p example implementation program in Debian. The line indicates the first upload to the archive on May 2. Not all users report popcon statistics so the numbers are a lower bound, and in reality are probably larger by a factor of 2 or 3.

Acknowledgements

- thanks to NSERC for its generous support of this research through its post-graduate scholarship program