Peer-to-Peer Downloading

CS 360 Internet Programming

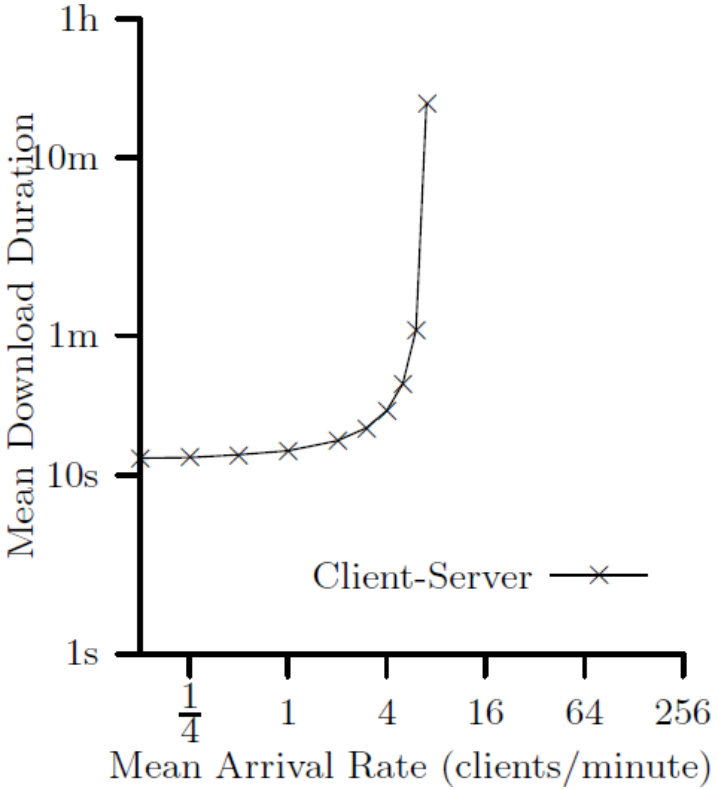
Daniel Zappala

Brigham Young University
Computer Science Department

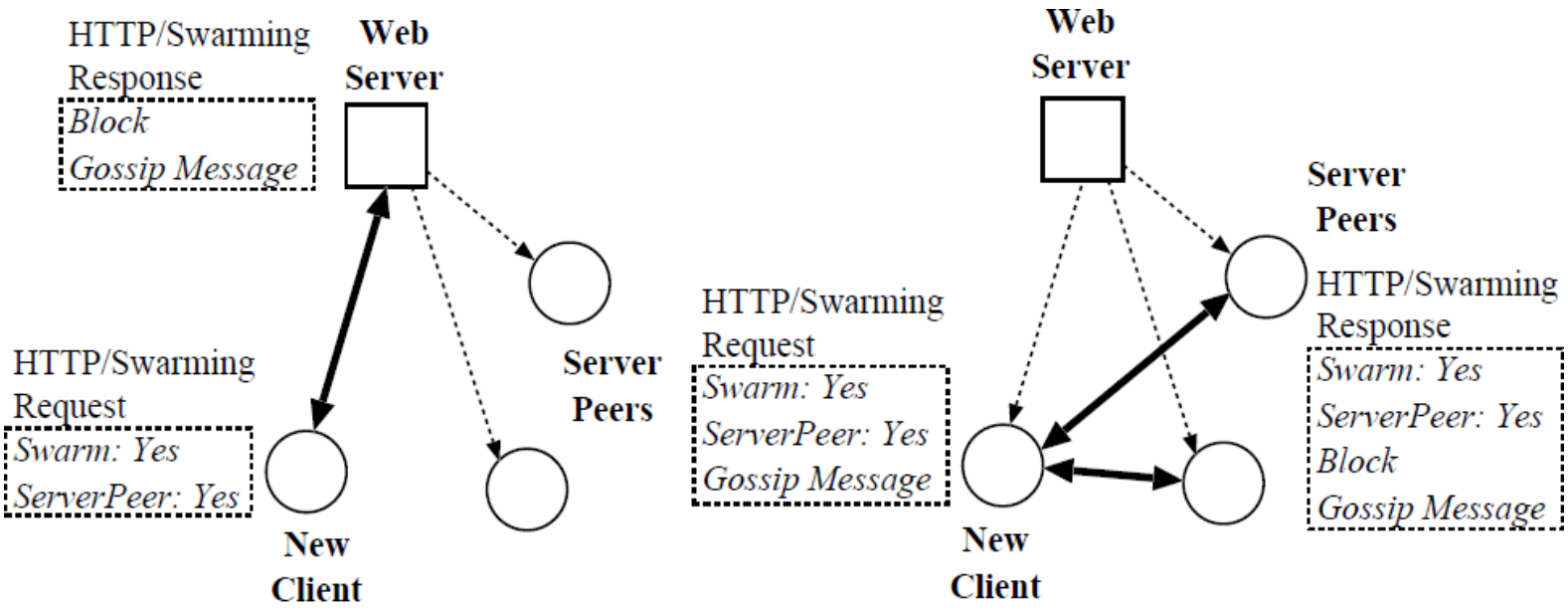
Motivation

- web servers are limited in their scalability
  - for a given web server, the more clients that need to be served, the slower they access the content
  - eventually the wait becomes so long, TCP connections time out
- CDN is a good solution, but expensive

serving a 1 MB file from a 1Mbps server



Swarming



(a) A new client gets a block and a gossip message.

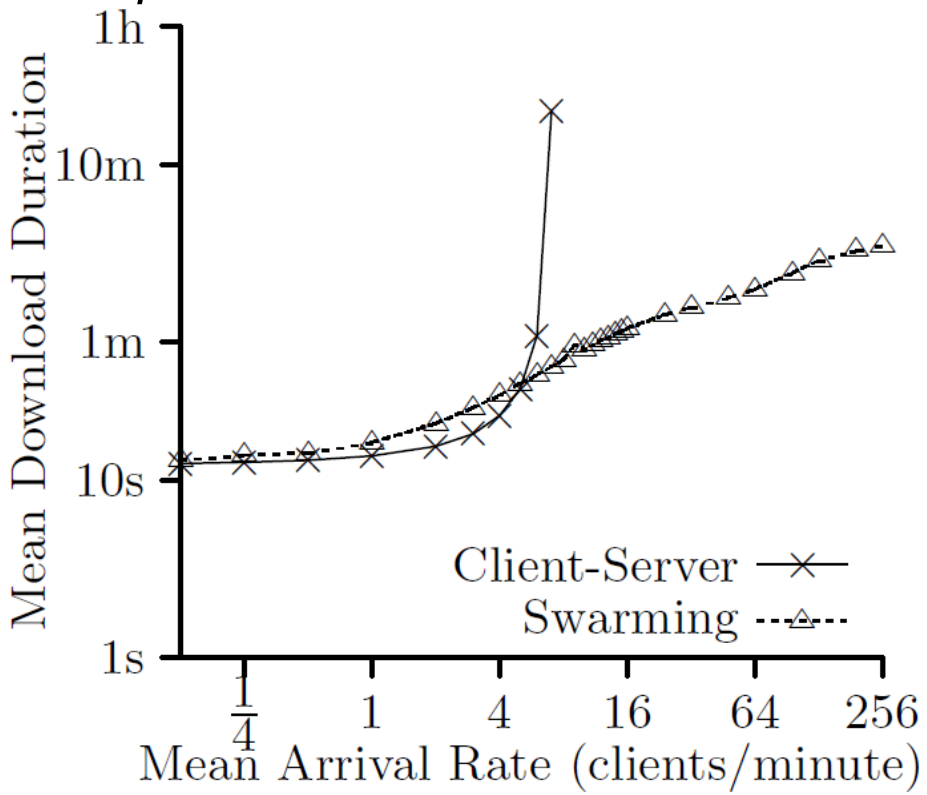
(b) The client finds and downloads blocks in parallel.

- peer-to-peer content delivery
  - serve blocks of a file to some peers
  - have these peers serve this content to other peers

Scalability

- self scaling: the more clients that participate – the more bandwidth is available

would need a 28 Mbps server to handle this load



Flash Crowds

- 6 clients/minute steady state
- client-server flash crowd: 1 hour of 12 clients/minute
- swarming flash crowd: 1 hour of 120 clients/minute

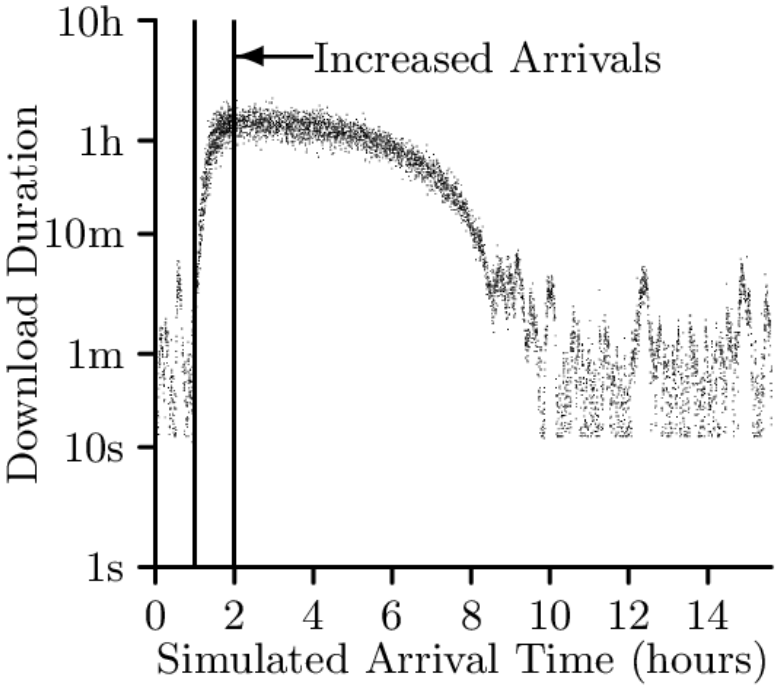


Fig. 6. Client-Server flash crowd

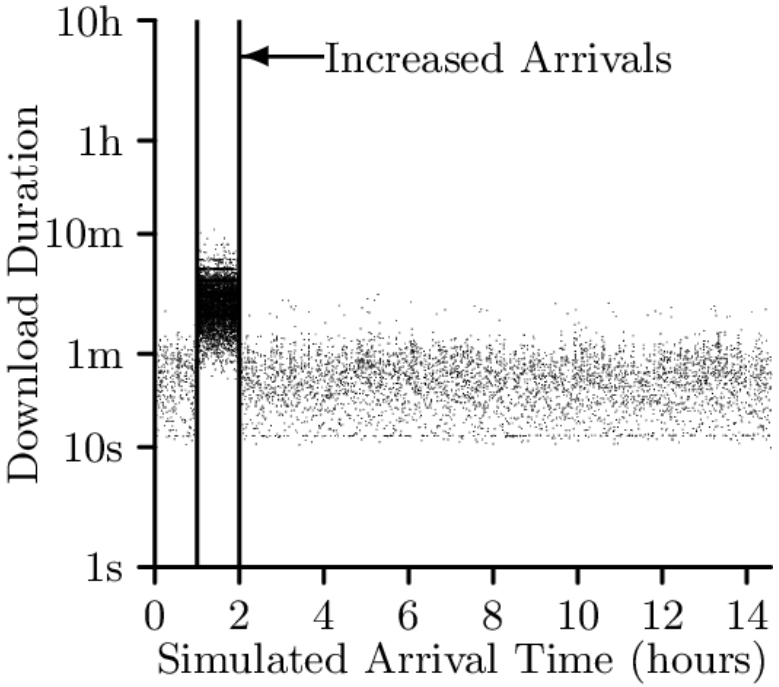


Fig. 7. Swarming flash crowd

Basic Mechanisms

- bootstrapping
  - download a .torrent file from a web server
  - contact the listed *tracker* for a list of peers
- peer discovery
  - periodically contact tracker if current peers don't have content you need or they leave the system
- content location
  - check with each peer to determine which blocks they have
  - download rarest blocks first

Content Delivery

- seed
  - web site must run a *seed* which has the entire file so that at least one peer is available that has the whole file
  - other peers that download the file may *linger* and act as a seed during this time
- parallel download
  - keep j connections active
  - download a block at a time from each connection
- incentives
  - serve content to k connections at a time
  - try to serve the connections that give you the most content
  - periodically serve content to a random connection to see if it can do better than a current connection

The .torrent File

- announce: the URL of the tracker
- info: a dictionary with the keys
  - name: suggested name for the file
  - piece length: number of bytes in each block of the file, default 256K
  - pieces: list of SHA-1 hashes of the blocks of the file
  - length: the length of the file in bytes
- other options for multi-file torrents

Communicating with the Tracker

- client sends an HTTP GET with:
  - SHA-1 hash of the info value from the .torrent file
  - peer identifier for this peer, chosen randomly
  - IP address of the peer
  - port number of the peer
  - number of bytes uploaded and downloaded so far
  - the number of bytes left to download
  - event: started, completed, or stopped
- tracker sends a response
  - interval: number of seconds the peer should wait before contacting the tracker again
  - list of peers that are active in the system, including peer identifier, IP address, and port

Connections

- peers make many TCP connections to other peers
  - when you finish downloading a block, tell your peers
  - you may choke (send no data) or unchoke (send data) each peer
  - you may tell each peer you are interested in their data or not interested
- if you are unchoked and interested, you may request blocks from that peer
  - always download from as many peers as possible
  - request blocks in random order

Choking

- two purposes
  - limit the number of uploading connections to provide good TCP performance
  - provide incentives - I will upload to you if you upload to me
- goals
  - cap number of uploads
  - avoid rapid switches between choking and unchoking
  - reciprocate to peers that let you download
  - try unused connections periodically to improve performance
- algorithm
  - operates once every 10 seconds
  - unchokes four peers for which you have the best download rates and who are interested in you (rolling 20 second average)
  - choose a single peer who gets unchoked regardless of its upload rate and who is not currently, rotate this peer every 30 seconds ,give new connections a better chance to be that peer
- seeds unchoke those peers with the best upload rates

Interesting Issues

- how well is it working?
  - what happens when people use it?
  - how well do incentives work?
- how can I make it work better?
  - what is the right peer selection strategy?
  - what is the right block selection strategy?