Swarms: First Class Citizens in the Future Internet

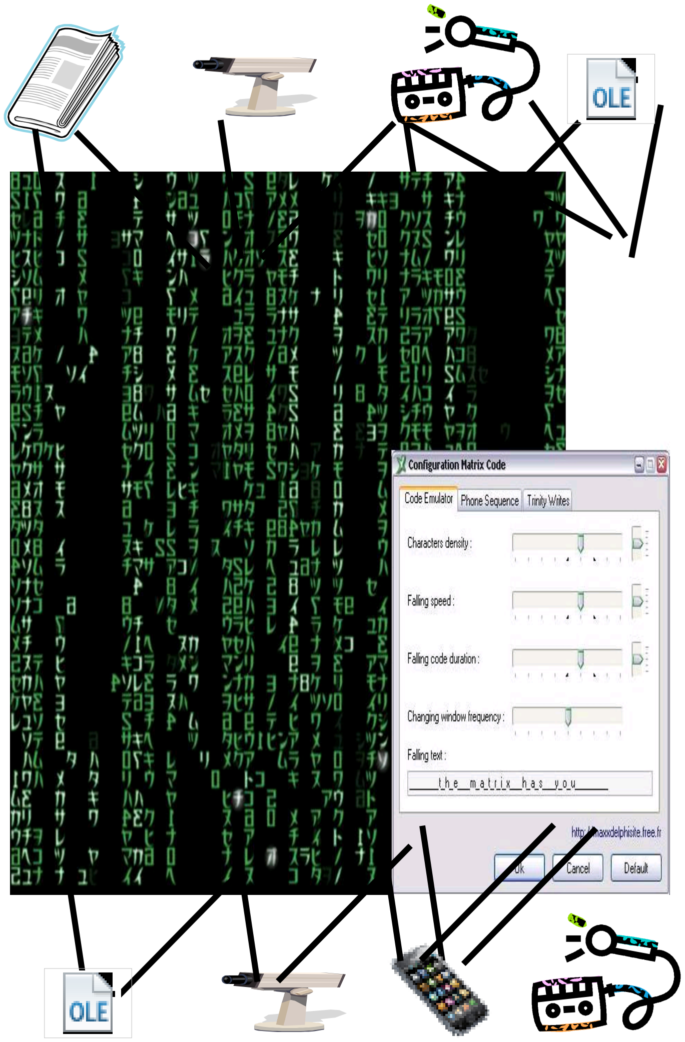
Don Towsley
UMass - Amherst

Vision

Information organized in one or more swarms

End hosts efficiently find, obtain information, content using simple local policies

Q: what should network provide?

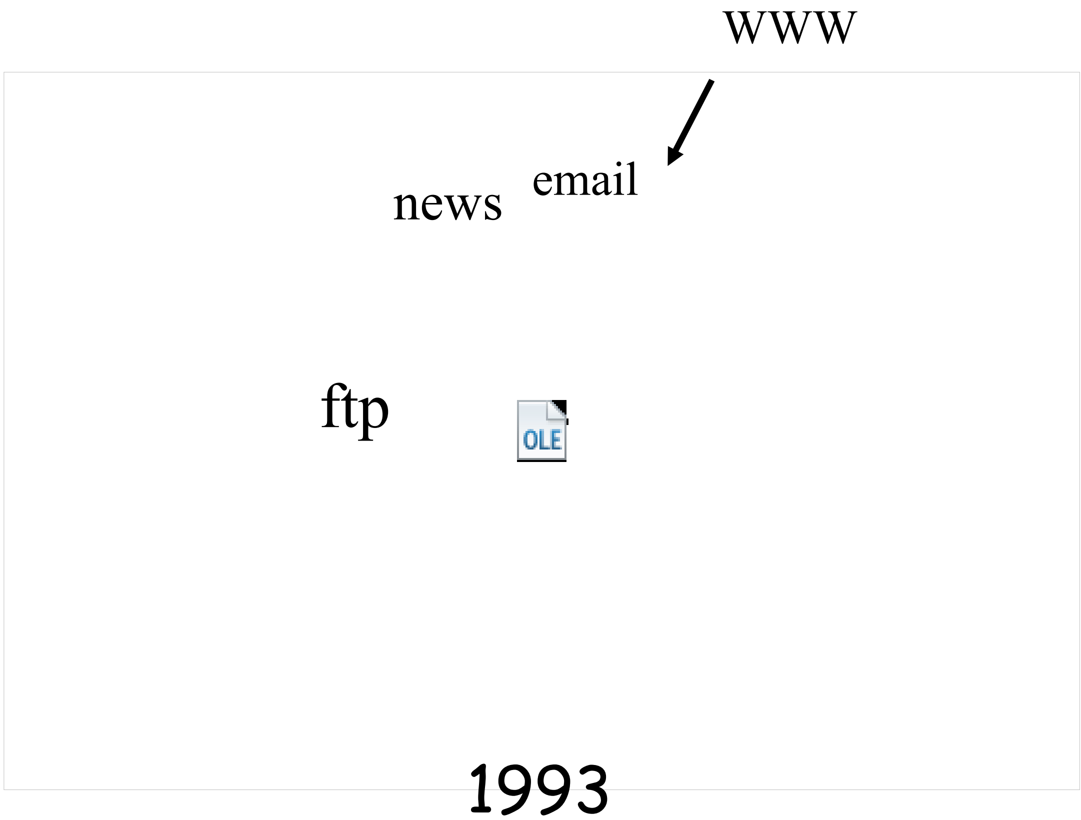


Outline

- a traffic-centric history of the Internet
- swarms
- an architecture to support swarms
- summary

Academic Internet: 1993

- 1993 - academic traffic



www

email

news

ftp

1993

The web: 1995

- 1993 - academic traffic
- 1995 - web appears

news ftp

news

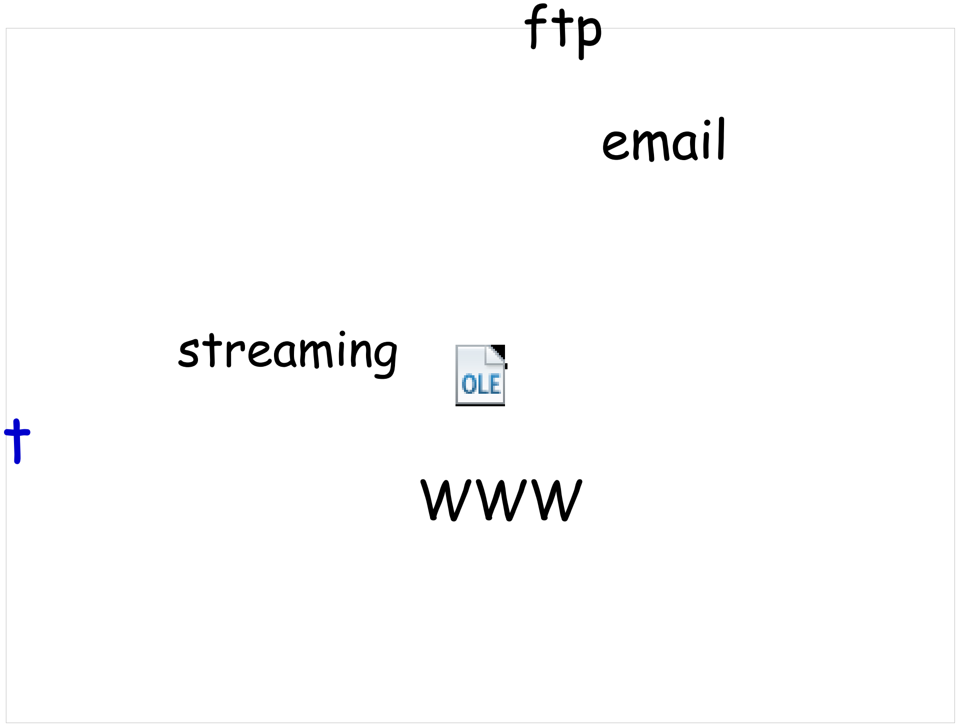
WWW email

1995

1993

Rise of the Web: 2000

- 1995 - web shows up
- 2000 - almost 75% web traffic
  - Napster, Gnutella just starting

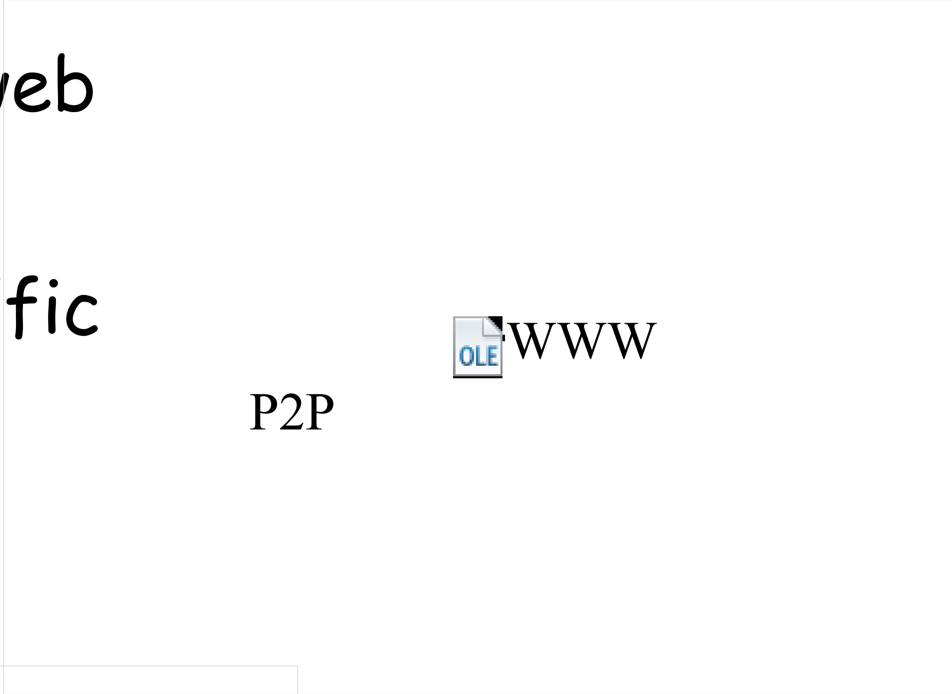


2000

1995

Rise of P2P: 2004

- 2000 - almost 75% web traffic
- 2004 - 67% P2P traffic
- over half BitTorrent (BT)
  - BT - canonical swarm

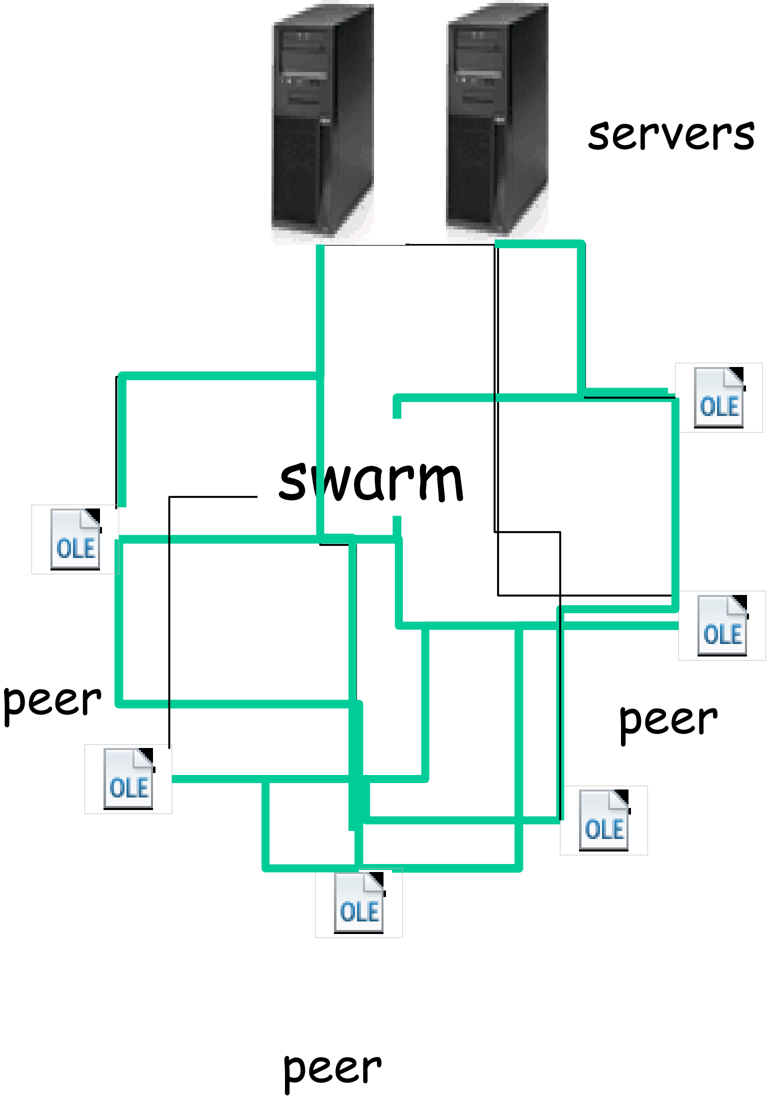


2004

2000

Swarm (BT) Tutorial (Cohen)

- large file (~GBs), divided into small pieces (256KB).
- peers initially given random set (50+) peers/servers
- peers fetch from 4 or more random peers/servers



Peer reselection

- peer connected to four peers at a time
- every 30 seconds
  - drops peer with lowest download rate
  - selects new peer (out of 50+) randomly

- multi-purpose mechanism
  - bootstraps new peers
  - connectivity
  - balances loads among peers ⇒ lower delays

Peer reselection

- peer connected to four peers at a time
- every 30 seconds
  - drops peer with lowest download rate
  - select new peer (out of 50-100) randomly

- multi-purpose mechanism
  - bootstraps new clients
  - connectivity
  - balances loads among peers ‖ lower delays
  - reduces need for traffic engineering

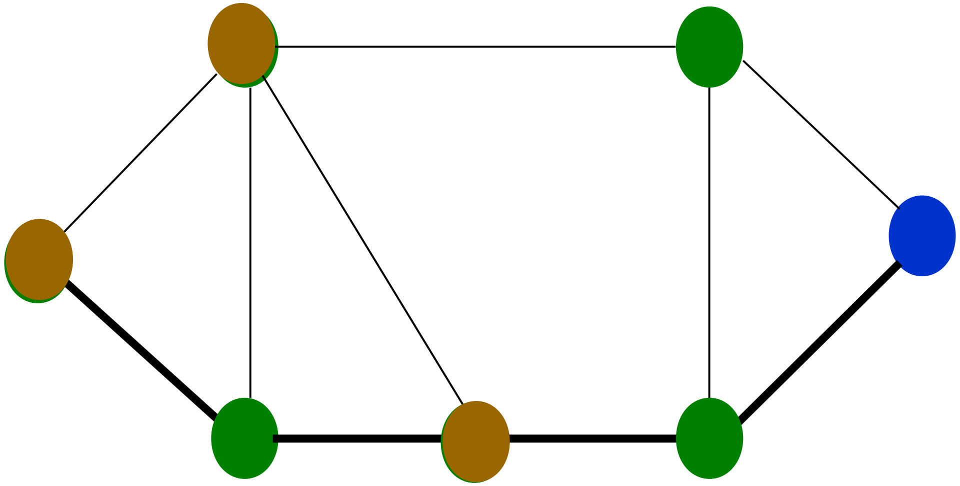
Ingredients to success of BT

- multiple connections / active downloads
- random peer reselection

Robust to failures, changes in traffic patterns

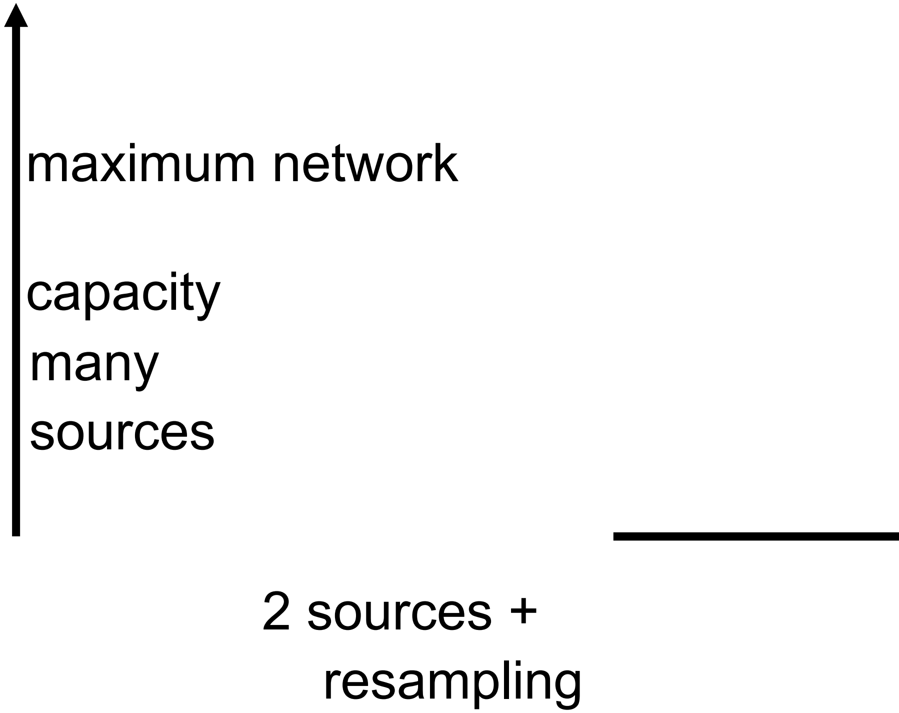
Multi-source data transport

- provide requestor “set” of sources
- requestor balances load across sources/ paths
- exhibits desirable load balancing properties
- BT transport mode
  - TCP (connection) rate



Benefits of multi-source transport

- capacity increases with number of sources per requestor
- two sources + random resampling achieves same capacity as using many sources
- BT uses 4 sources + resampling



maximum network capacity

many sources

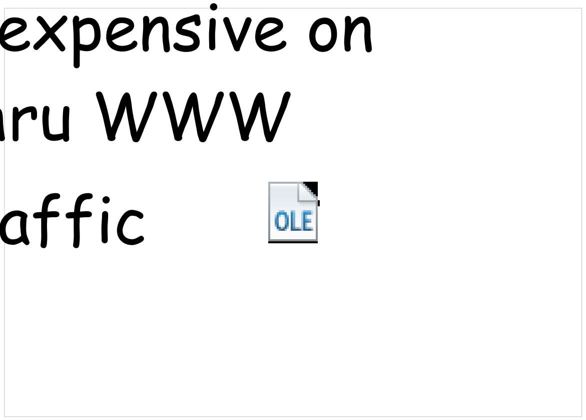
2 sources + resampling

Rise of on demand video: 2007

- P2P traffic shrinks from 67% to 37%
- fraction of swarm traffic swarm fraction shrinks

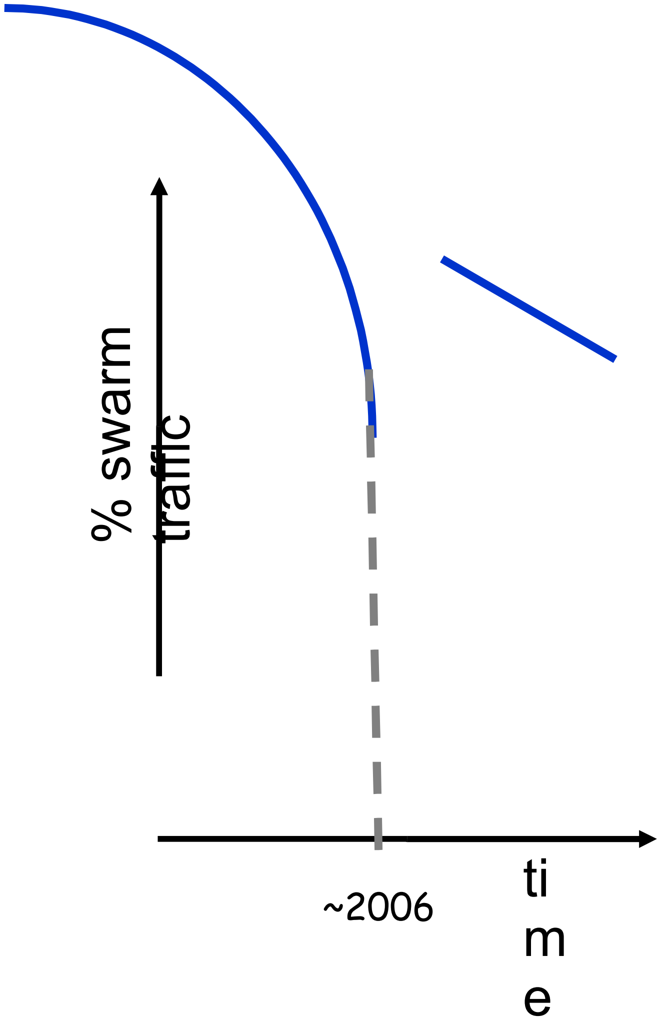
Why?

- availability of inexpensive on demand video thru WWW
- 10% YouTube traffic



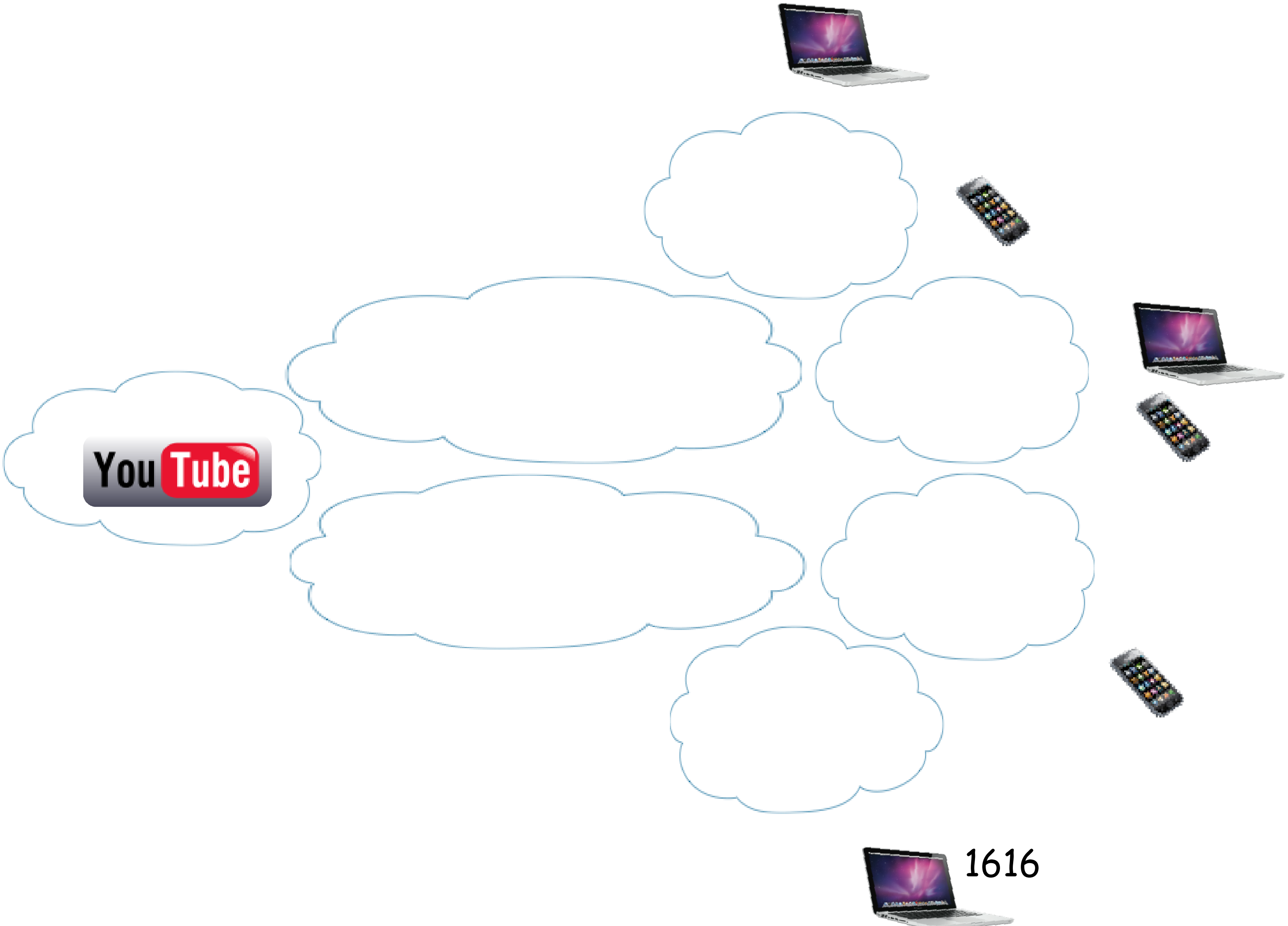
Current trends

- swarm traffic growing in absolute terms
- shrinking in relative terms
- rest of traffic single source, path
- trend more prominent in developed countries



How can we reap the benefits of swarms?

YouTube model



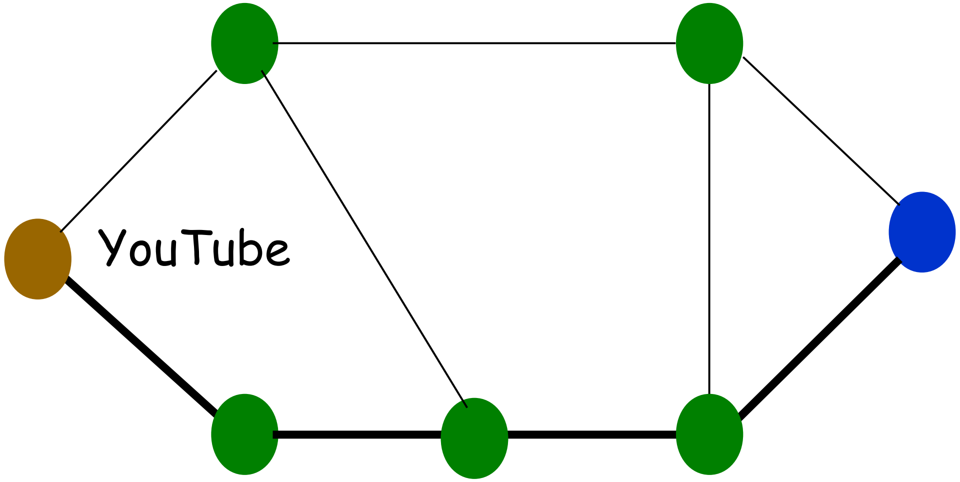
Can YouTube benefit from swarming?

Click to edit Master subtitle style

Without introducing BT?

Joint routing/rate control

Consider source-destination pair

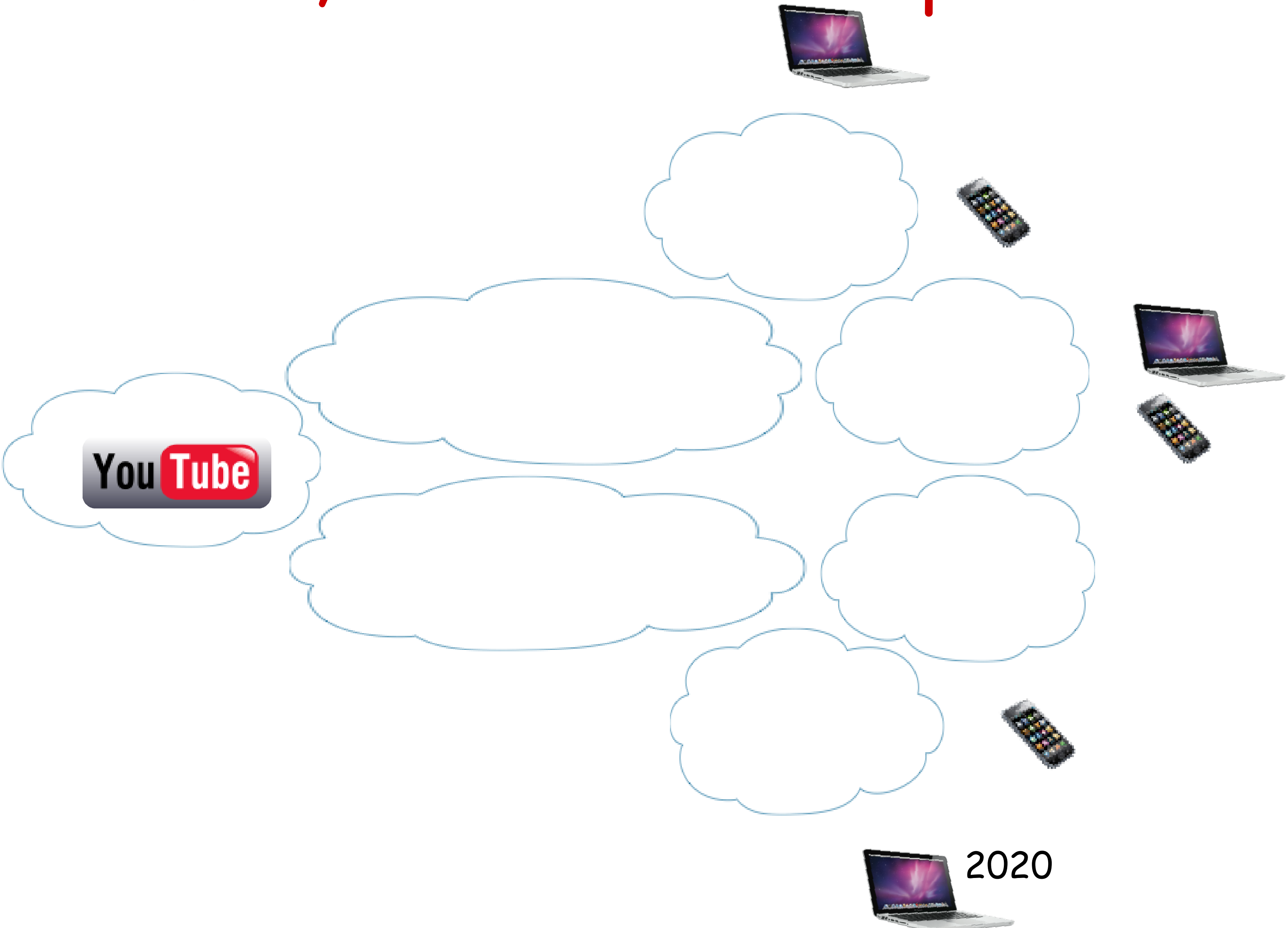


- provide session "set" of paths
- session balances load across paths
- well studied within Kelly optimization-based framework

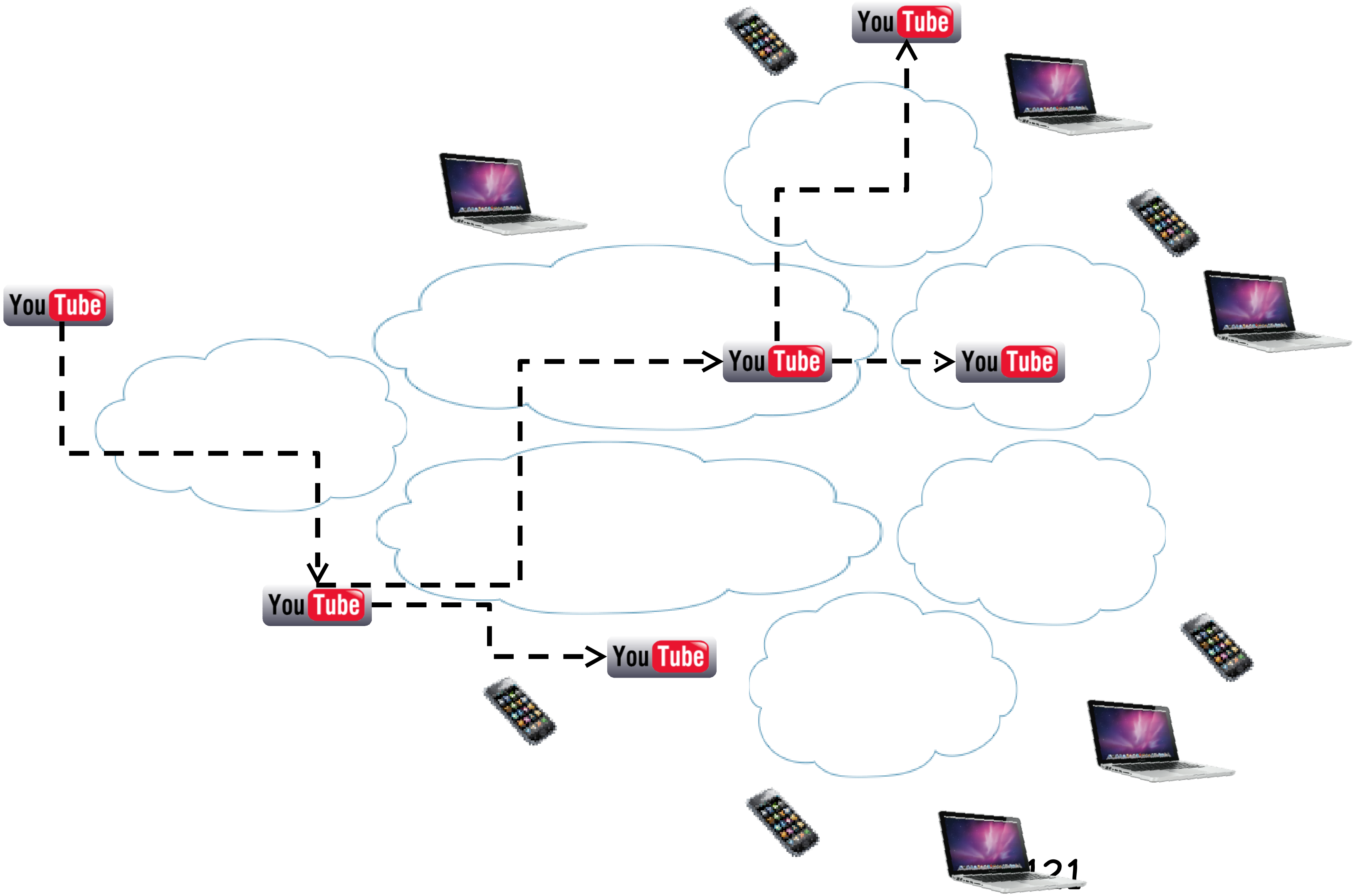
Joint rate/route control

- similar load balancing properties to multisource
- increasing no. paths increases capacity, robustness
- two paths with random replacement of worst as good as using all paths

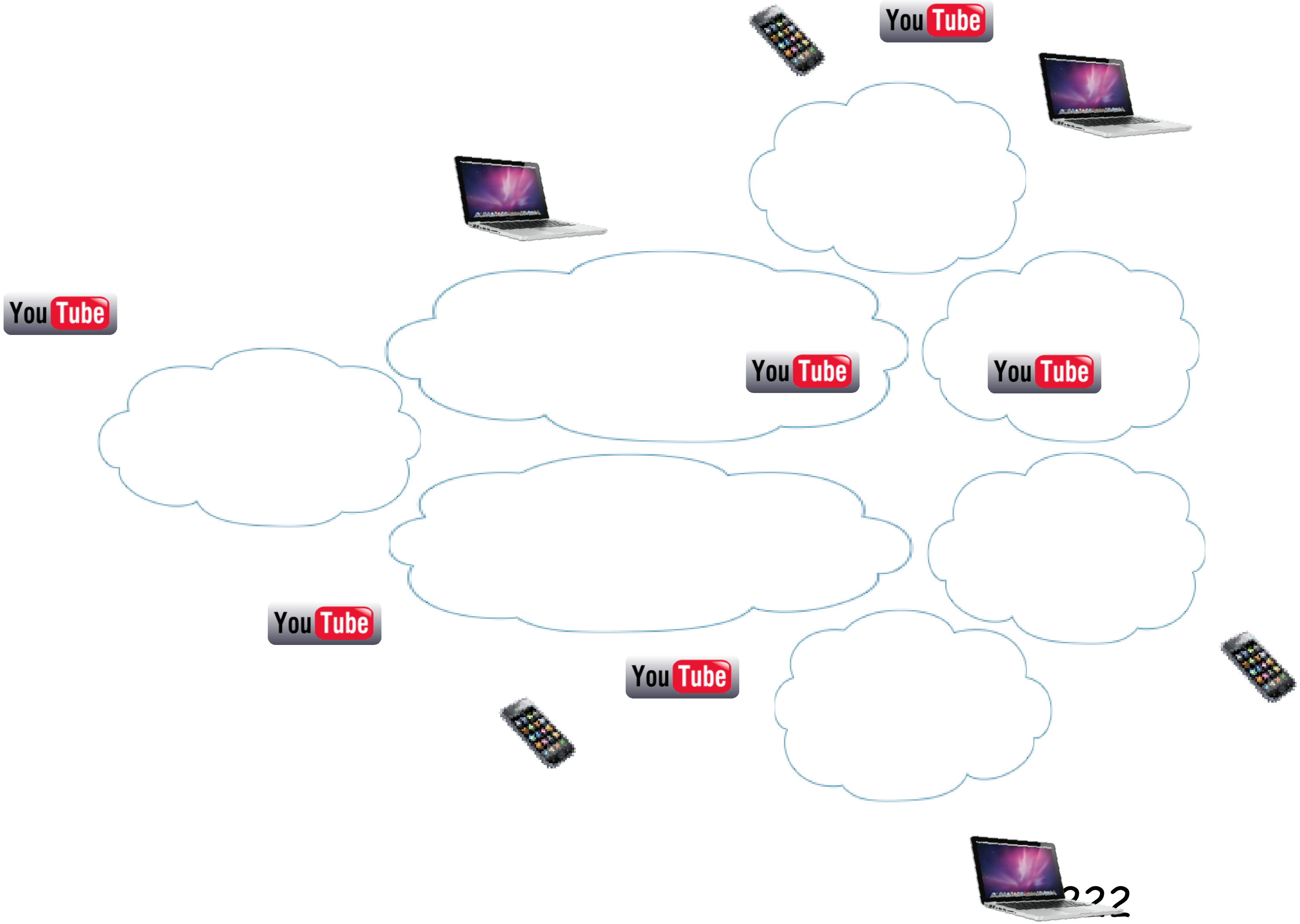
Netflix, YouTube: multipath



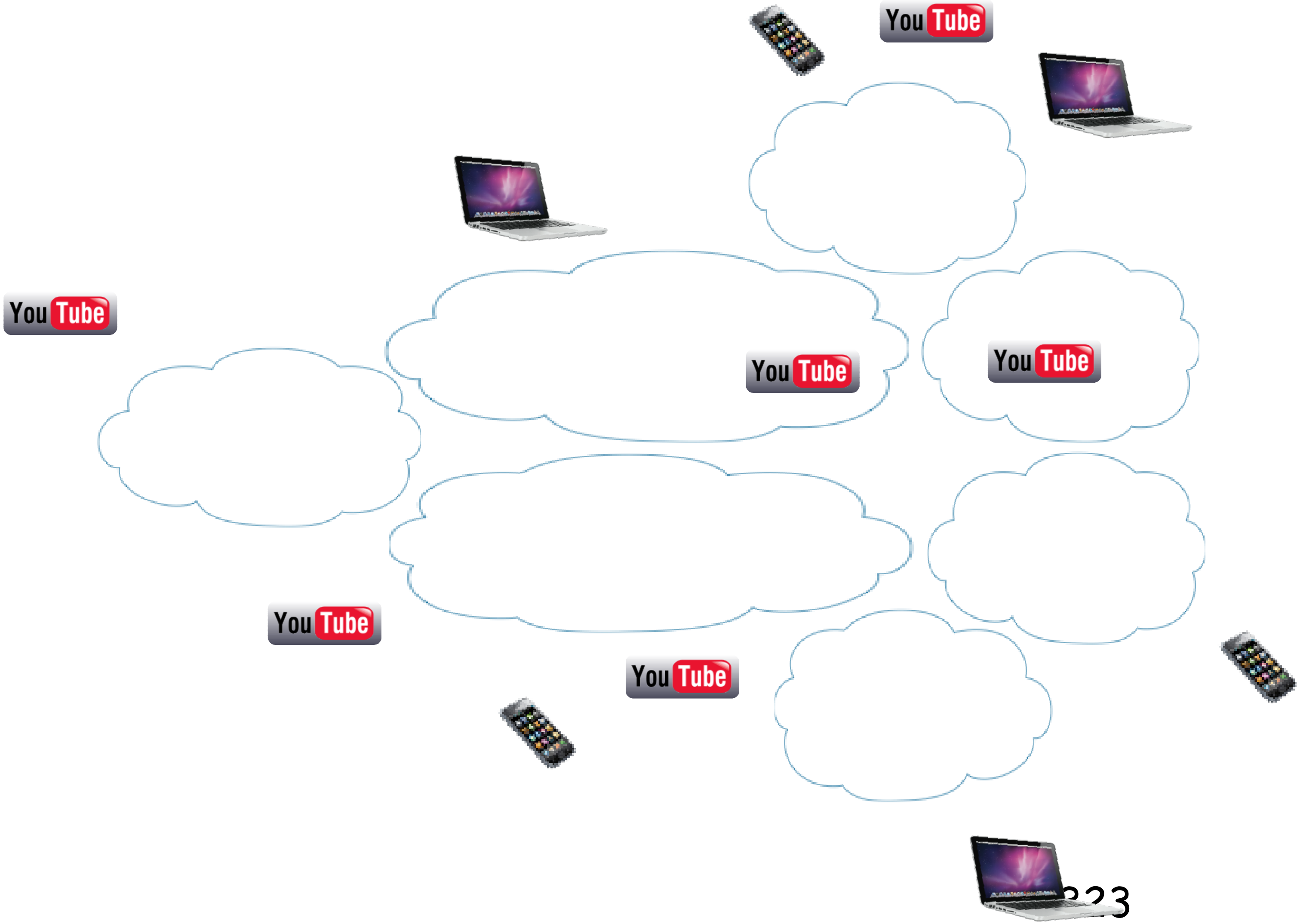
but YouTube replicates



can add multi-source

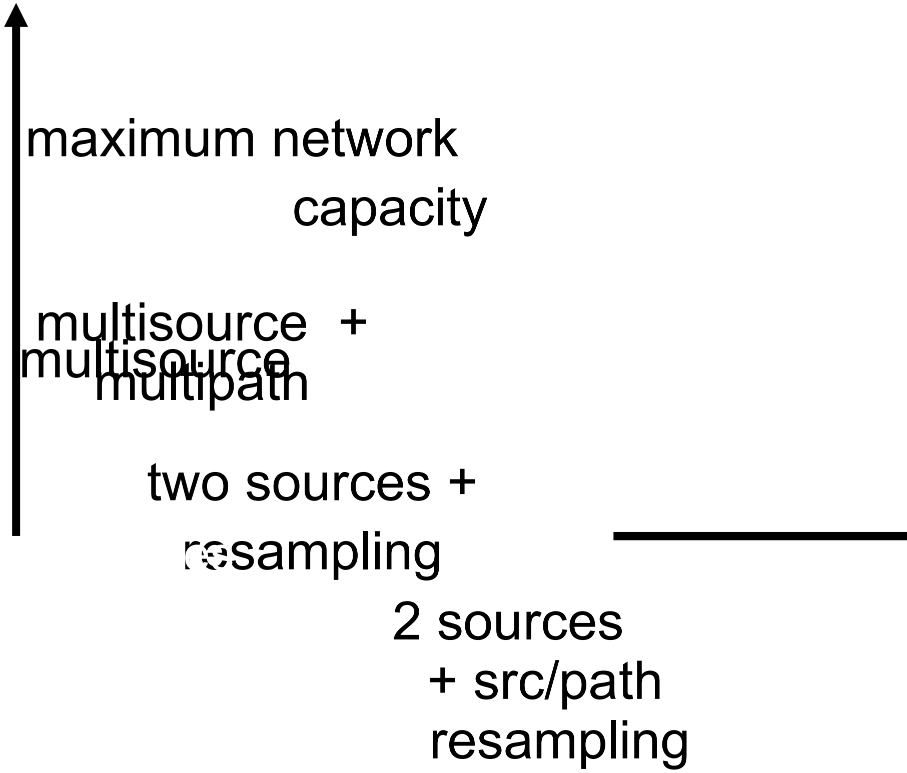


and multi-path



Multisource + multipath

- ❒ capacity increases with number of
  - ❍ paths per session
  - ❍ sources per session
- ❒ two sources + random source/path resampling achieves same capacity as many sources, paths
- ❒ high tolerance to



Issues

- controller design
- path/source selection
- scheduling
- locating content
- where, when, what to replicate
- security

Controller design

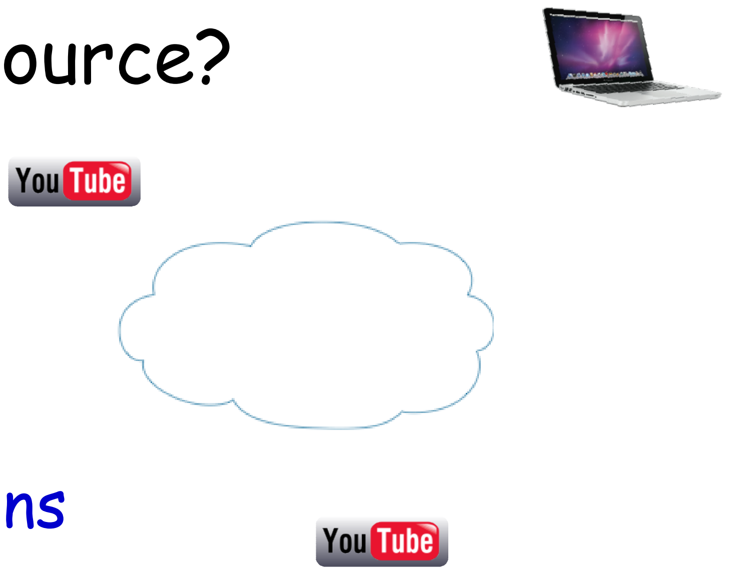
- falls within Kelly optimization-based framework
- many designs
  - different fairness criteria
  - convergence, stability, …
- receiver oriented design
- IETF efforts on multipath rate control

Source/path selection

- sources
  - tied to finding content - defer to later
- paths
  - multihoming, increasingly common
  - supported by IPv6

Scheduling

Q: what to obtain from which source?



A: coding –

- divide content into chunks
- fetch random linear combinations of chunks from any source
- chunks solve set of linear equations
- combinations precoded, coded on the fly

❑ no coordination required

Naming, addressing

- name is the address (Jacobsen)
  - Named Data Networking (NDN) project (Zhang, Jacobsen)
- avoids need for name-address translation
- handles mobility
- hierachical names
  - (e.g., levels of interest, replication)
- route setup ü content search

Routing functionality (Jacobsen)

content storage

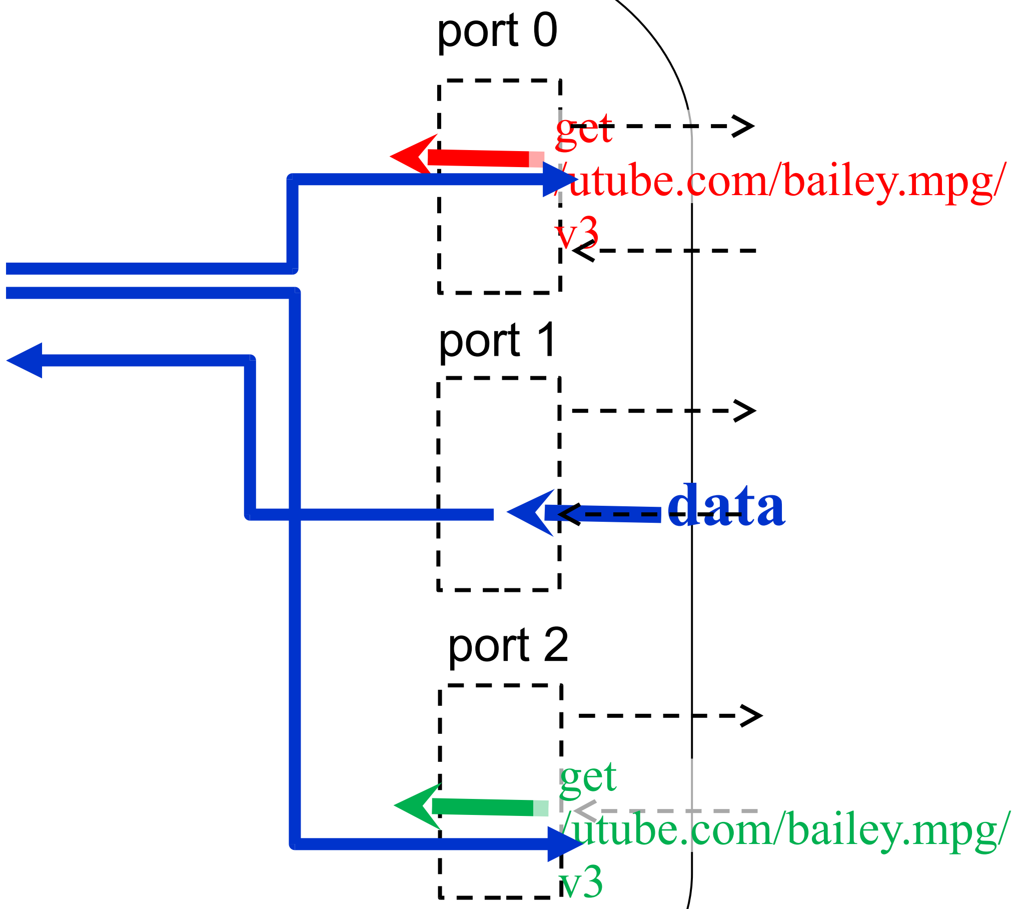
name	content
/utube.com/bailey.mpg/v3 data	

pending search table

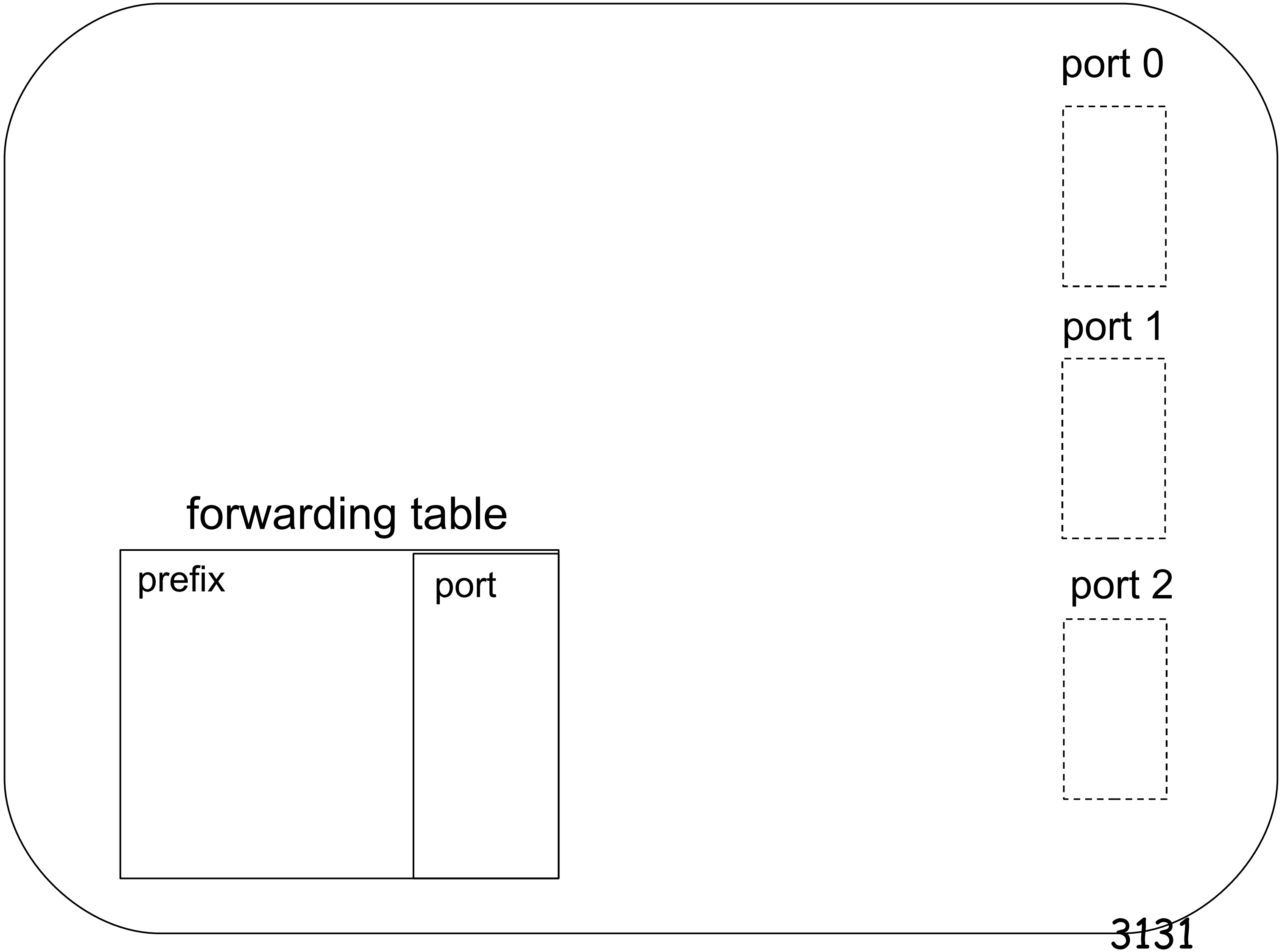
prefix	port
/utube.com/bailey.mpg/v3 0	2

forwarding table

prefix	port
/utube.com/ 0,1	



IP router



forwarding table

prefix	port

Finding content

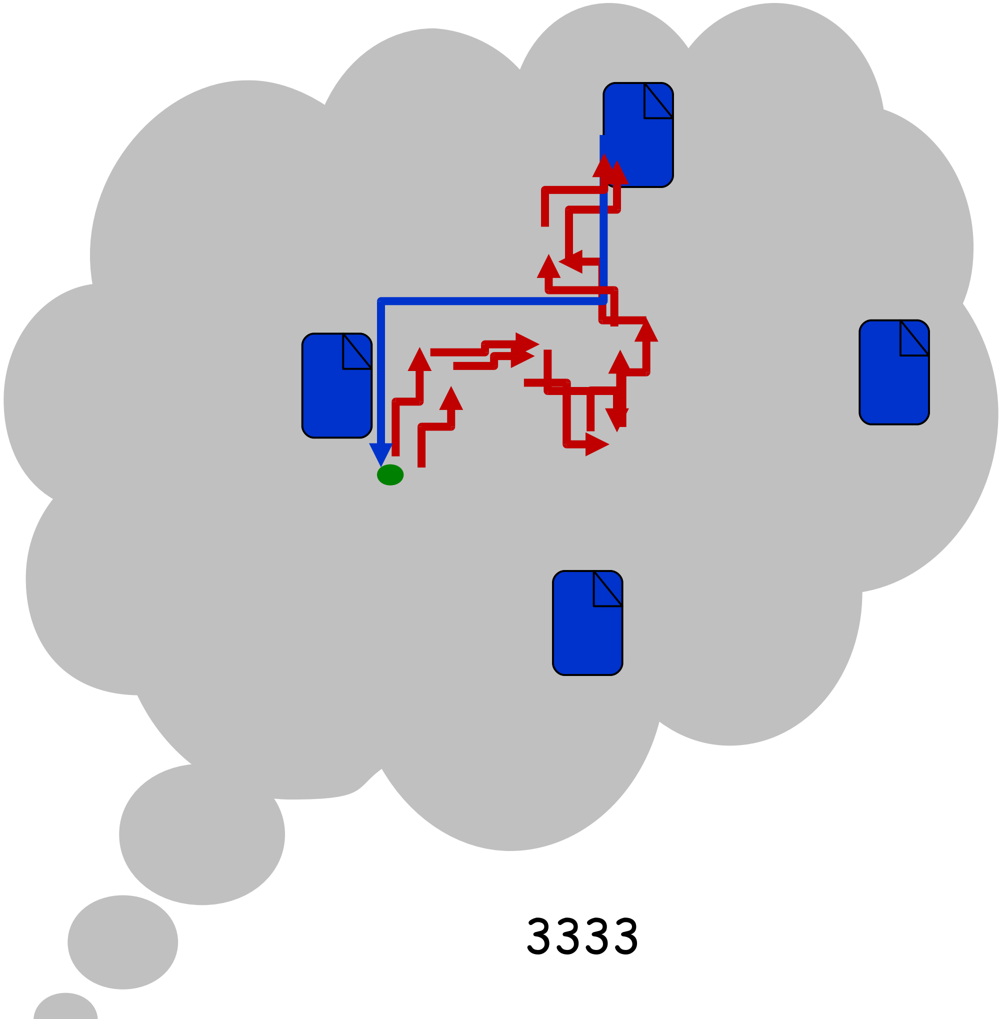
- simple “swarm-like” techniques
    - random walks
    - expanding ring search
- third party search engines
    - google
    - bing

Random walk search

- propagate query randomly through network
- halt when found, or hit max. TTL

Scalability in network size:

- O(1) communication overhead
- low delays



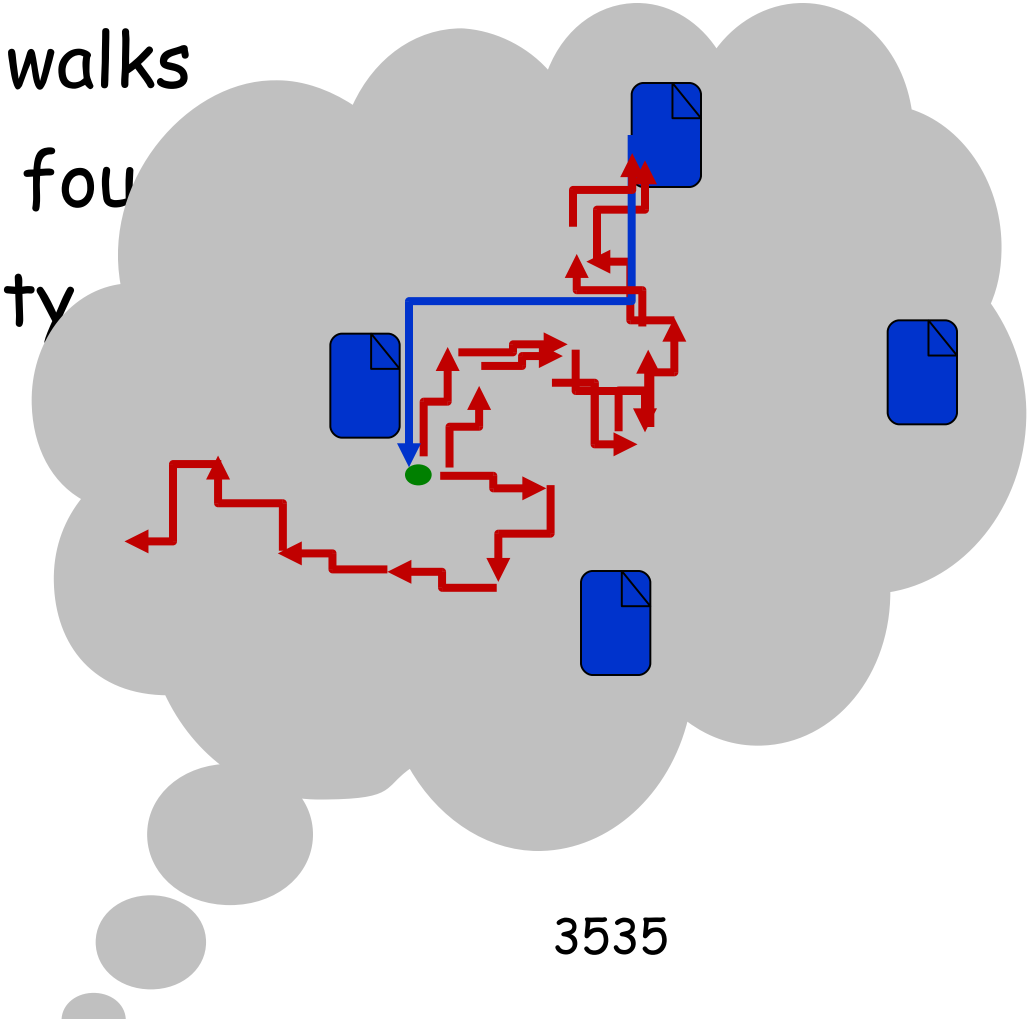
Random walk performance

	Popular	Unpopular
Many replicas	overhead O(1), search time O(1)	overhead O(1), search time O(1)
Few replicas	overhead O(n), search time O(n)	overhead O(1), search time O(n)

Long delays, high overhead for less replicated content

Multiple RW search

- initiate *k* random walks
- halt once content fou
- tune *k* to popularity



Multiple random walk performance

	Popular	Unpopular
Many replicas	overhead O(1), search time O(1)	overhead O(1), search time O(1)
Few replicas	overhead O(n), search time O(log2 n)	overhead O(1), search time O(log2 n)

How do we reduce overhead?

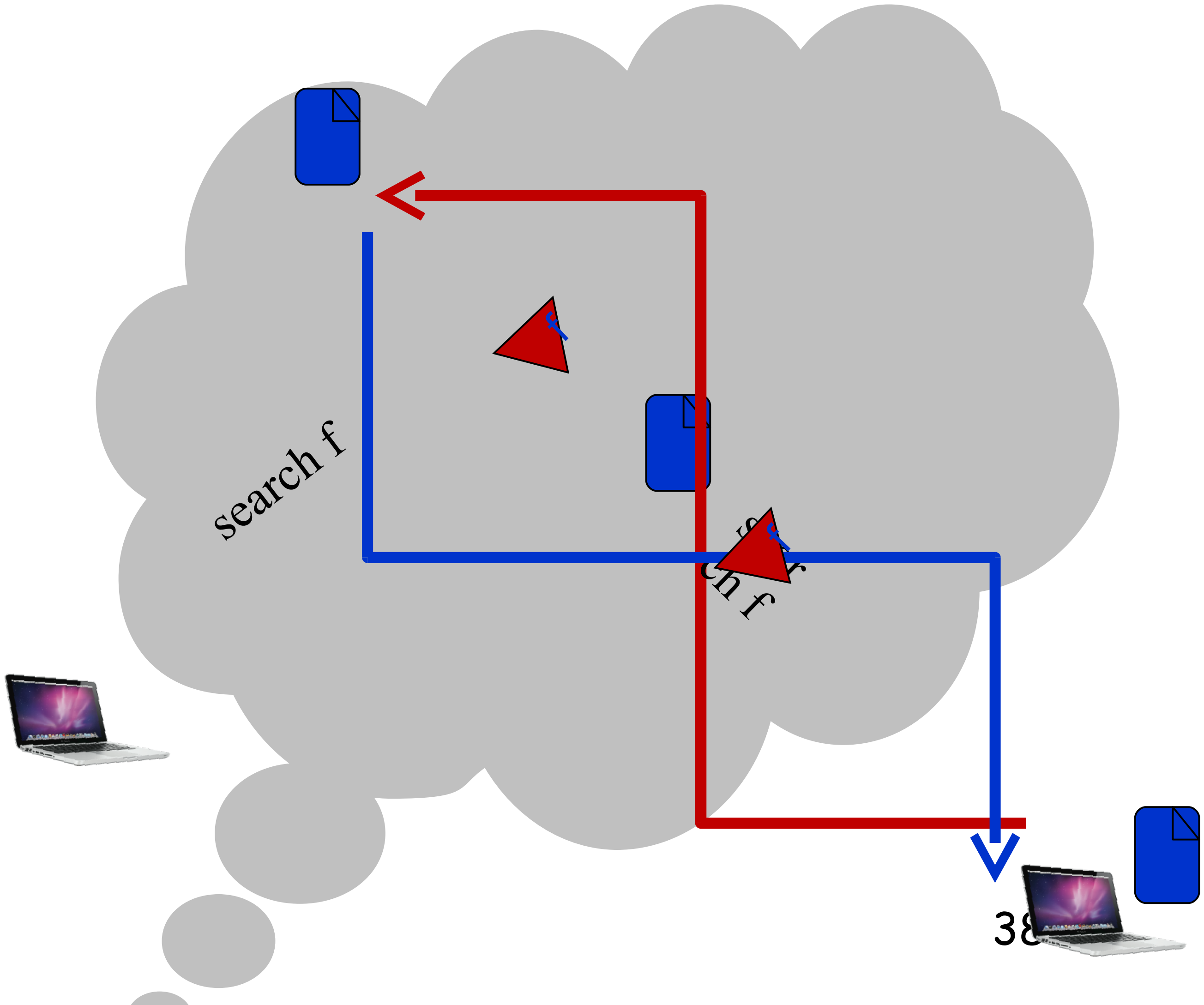
Search times

- couple no. copies to popularity
  - popular content ü   many copies

and/or

- introduce breadcrumbs
  - entries in forwarding table

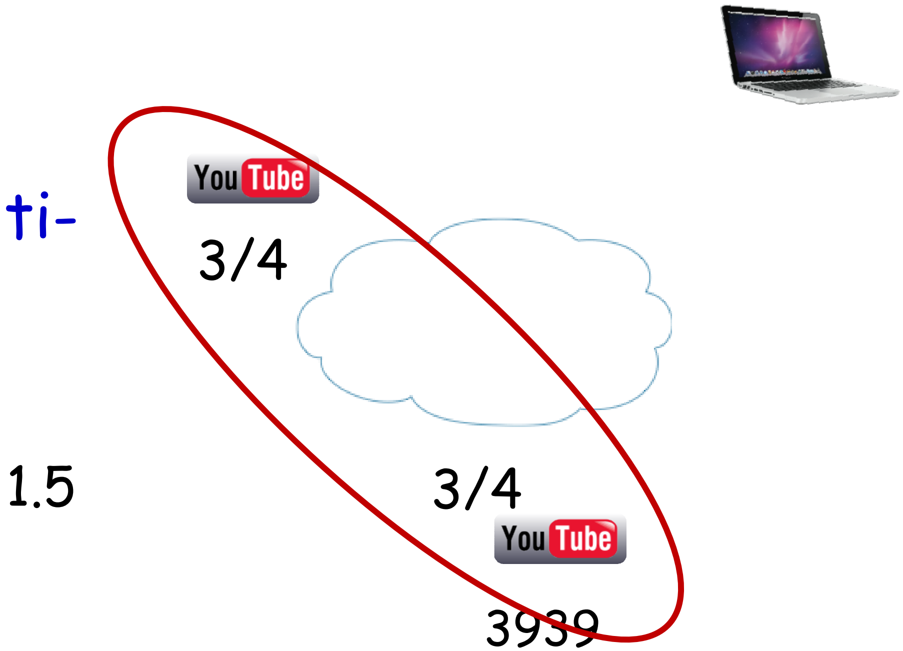
Breadcrumbs



Search: research questions

Replication

- when?
  - always? randomly?
- partial replication?
  - random coding + multi-source permits this



Content search: summary

- swarming techniques
  - well defined searches
  - handles most searches
- Google model
  - complex searches
  - added value searches
  - third party ($$$)
- hybrids?

Security

- secure content - not connection
- convert randomization into security

  "dynamic secrets"
- code so that requestor must obtain pieces from a minimum number sources

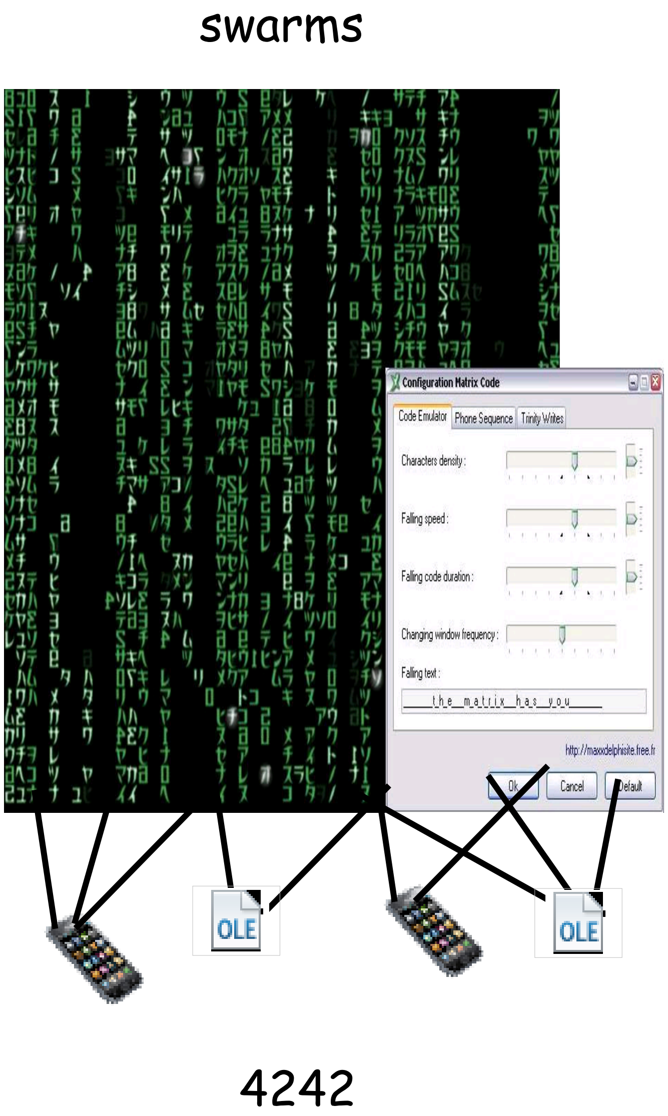
A network swarm architecture

End hosts

- network control
- simple, robust, local randomized policies

Networks

- rich connectivity
- content replication
- sets of paths as needed



Falling text : the_matrix_has_you

Parting thoughts

- swarms simplify network management/ control
  - network responsibility: rich connectivity
  - infrastructure to support swarms
- applies to
  - real-time, multimedia
  - mobile, wireless

Parting thoughts

- details to work out
  - technology for swarms
  - infrastructure to support swarms
  - content replication, searching
  - coding
  - security
- economics of swarms
- rich set of PhD research problems

considerable latitude for architecture and

The end

Thanks and questions

Slides (soon) available at

http://gaia.cs.umass.edu/towsley/INRIA-11.pdf