

Peer-to-Peer networks

Apu Kapadia

Assistant Professor
Computer Science, Informatics

Oct 3, 2011

Sharing a large file

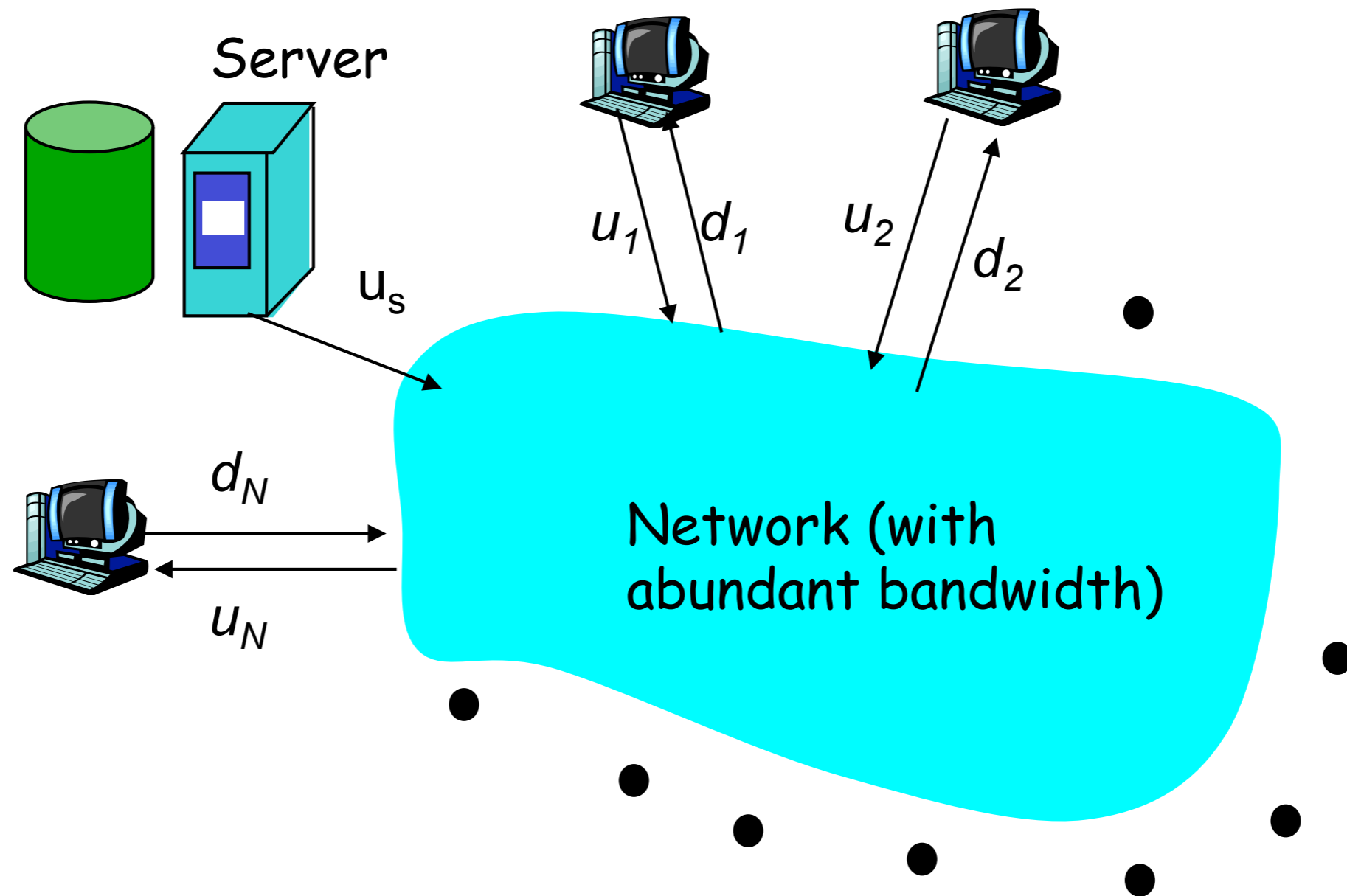


Image taken from Kurose, Ross slide set

Sharing a large file

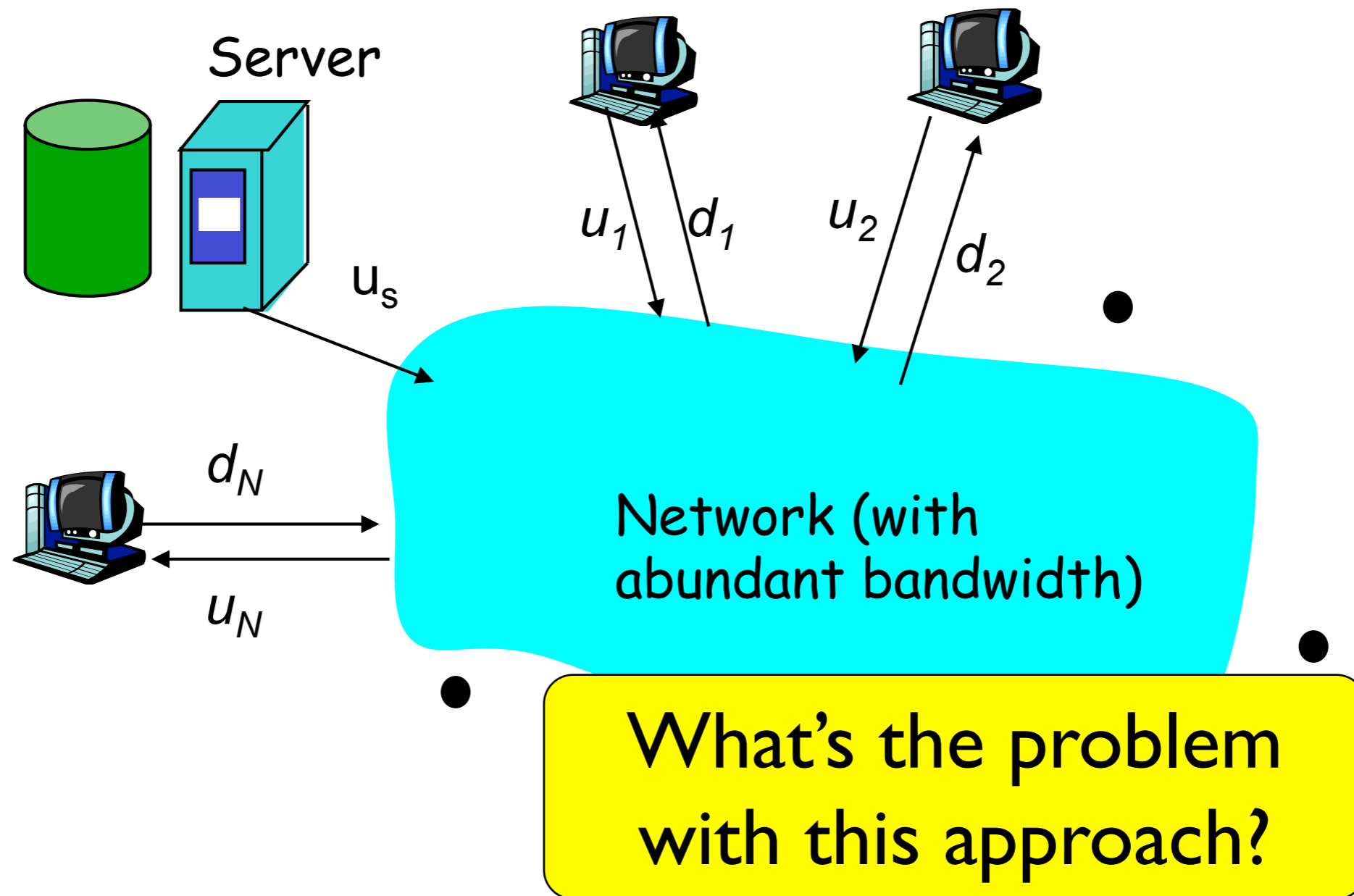
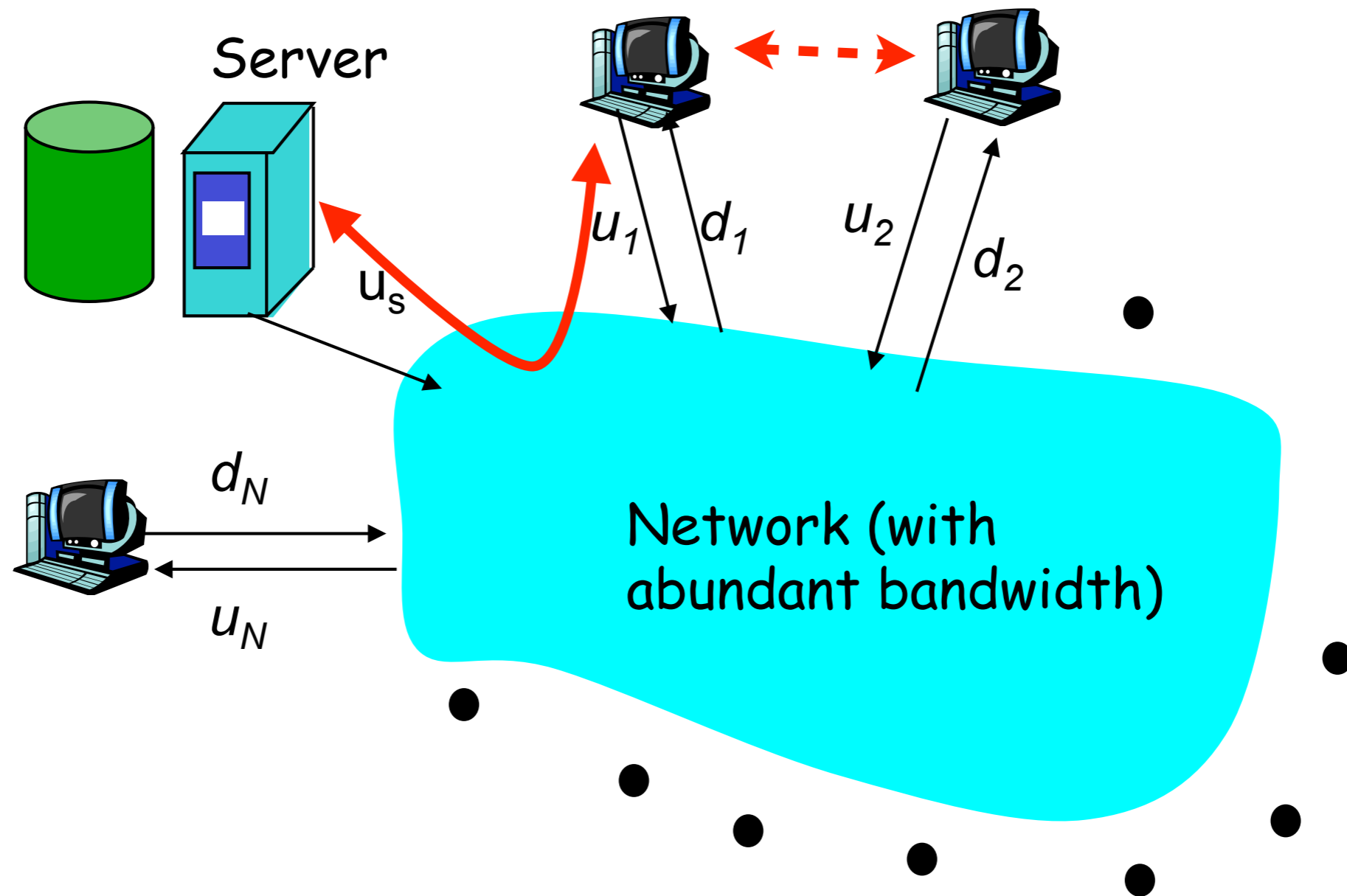
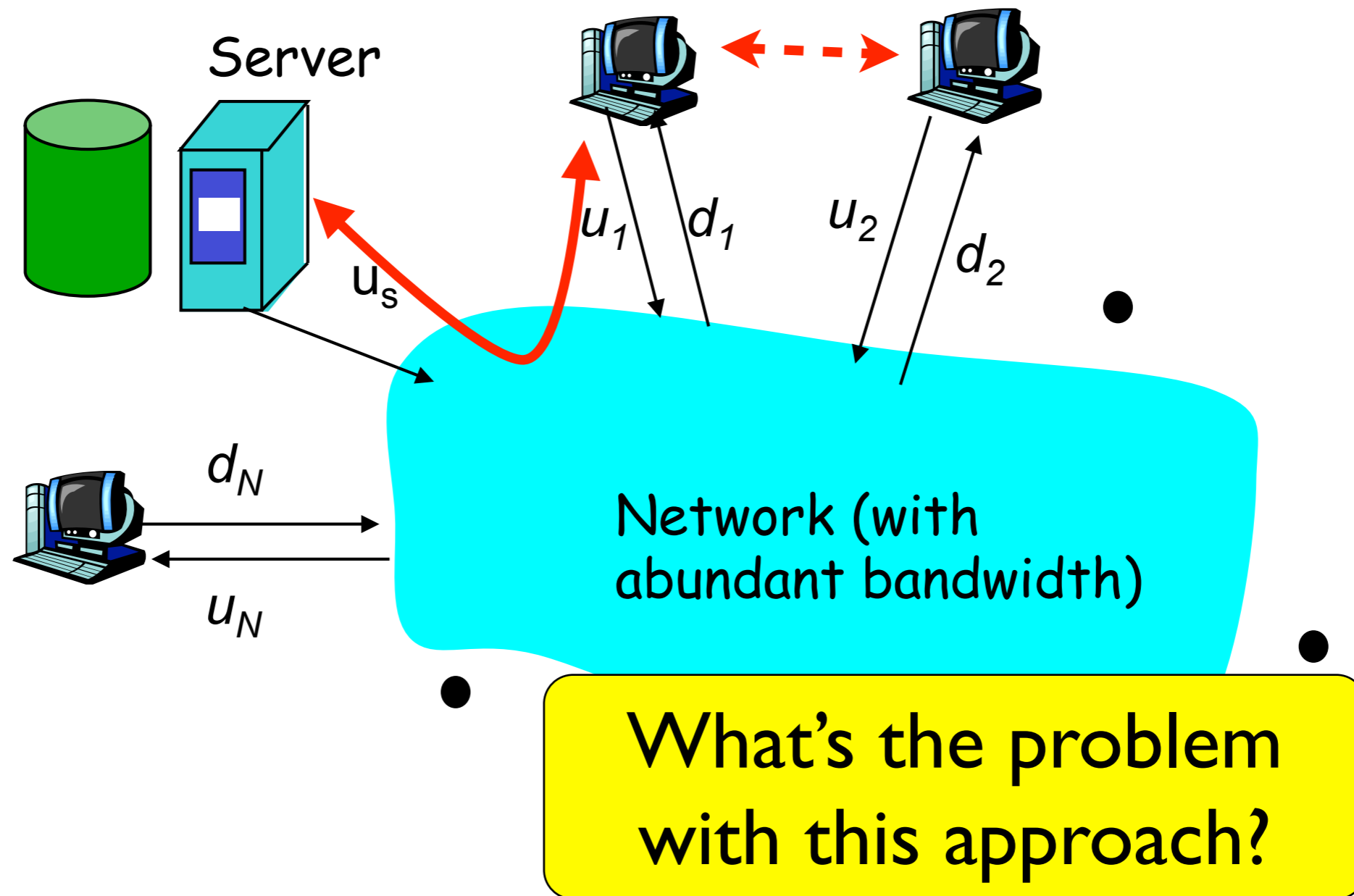


Image taken from Kurose, Ross slide set

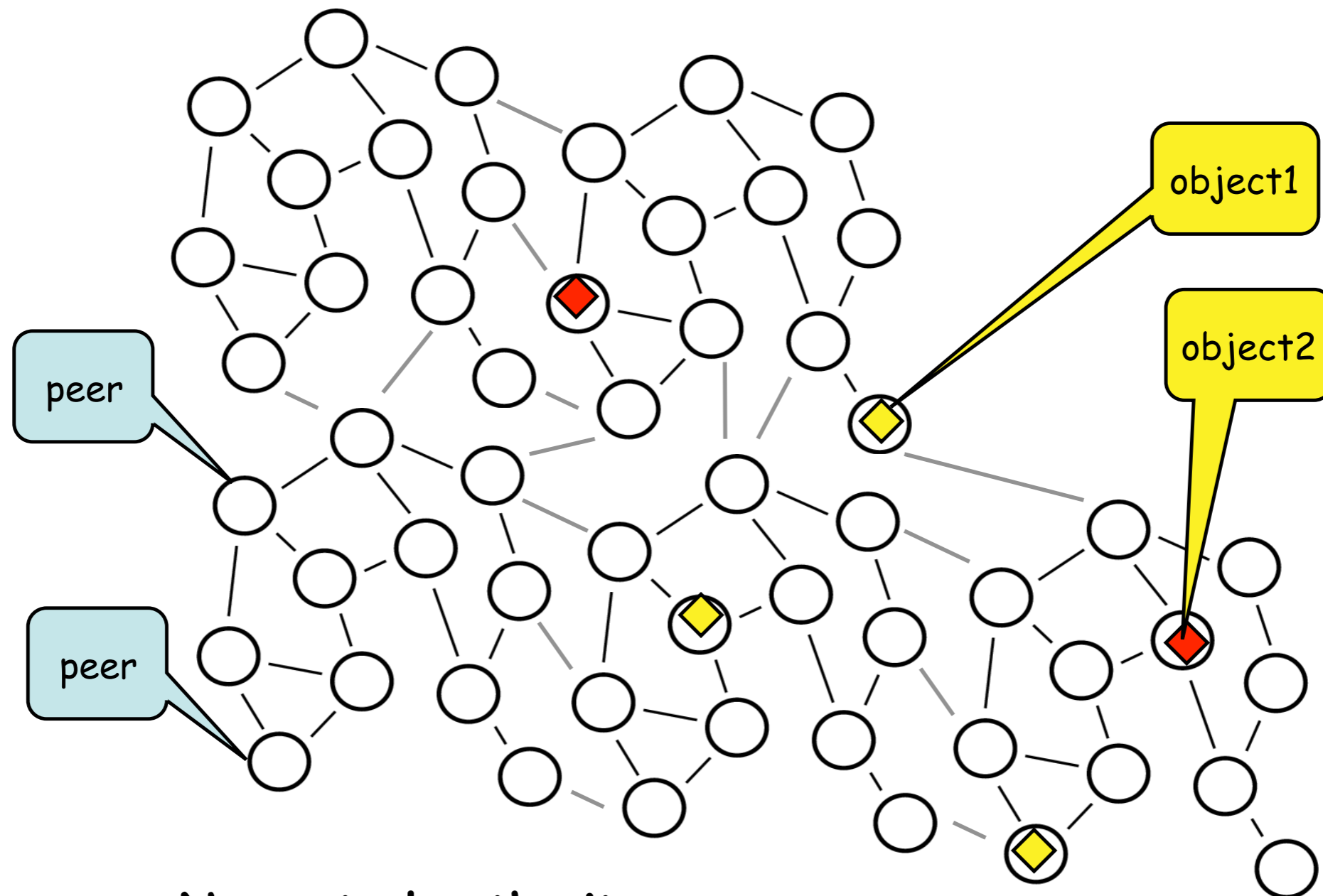
Napster: Server acts as a directory



Napster: Server acts as a directory



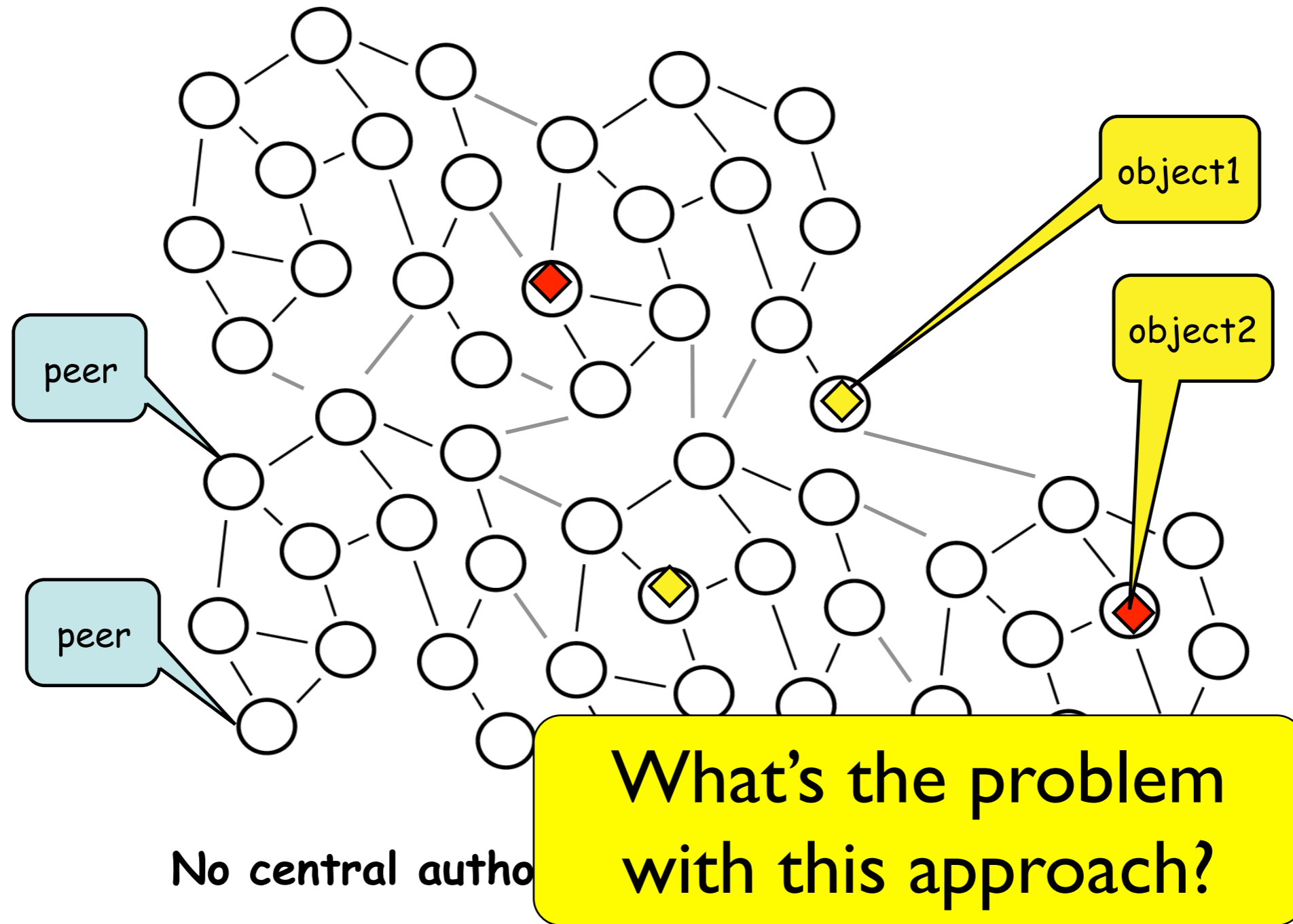
Gnutella: Get rid of the central directory



No central authority.

Image taken from Sukumar Ghosh's slides

Gnutella: Get rid of the central directory

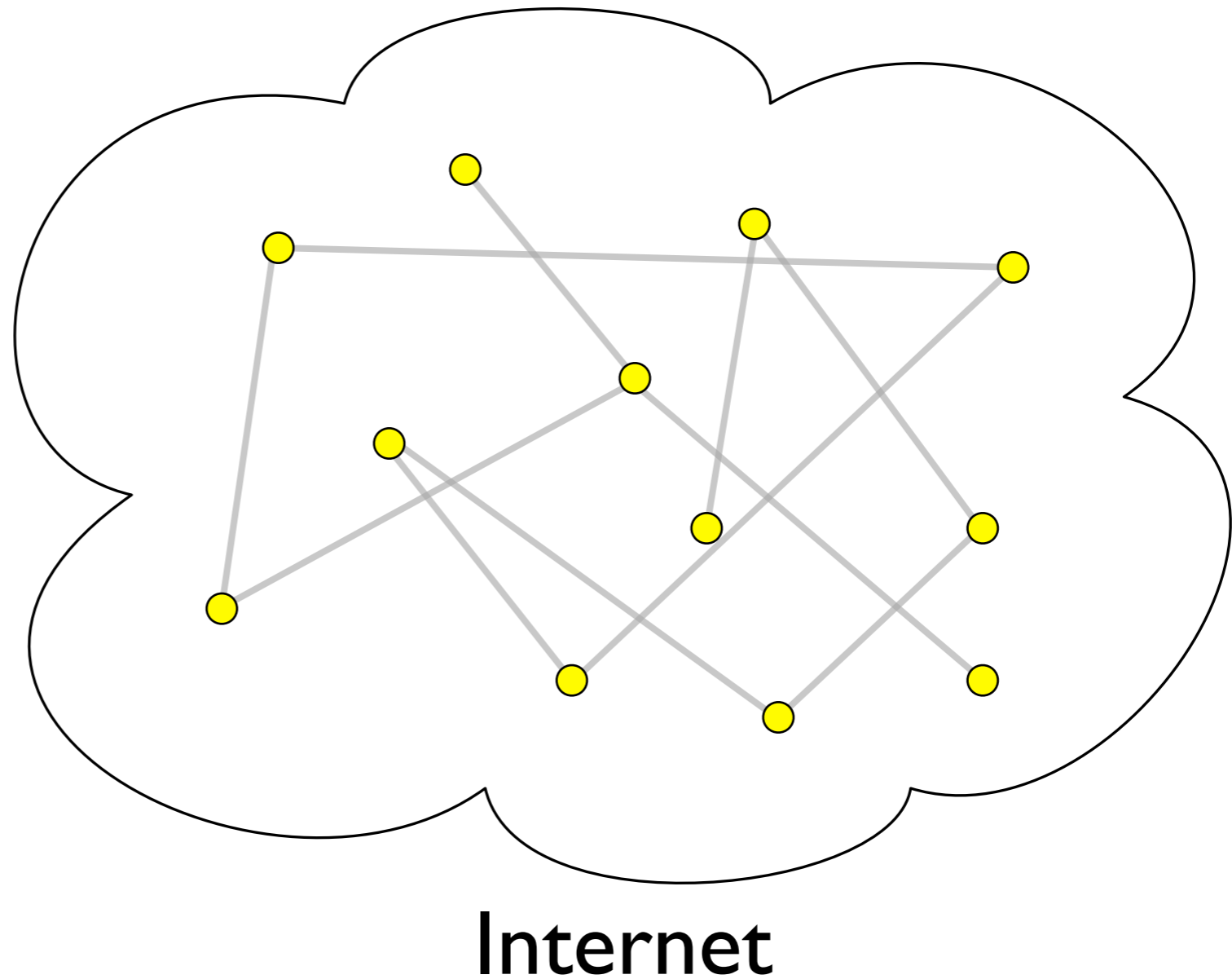


Two major problems with Gnutella

- Flooding is wasteful
 - lots of messages sent during searches
- Freeriding
 - peers may download but not upload

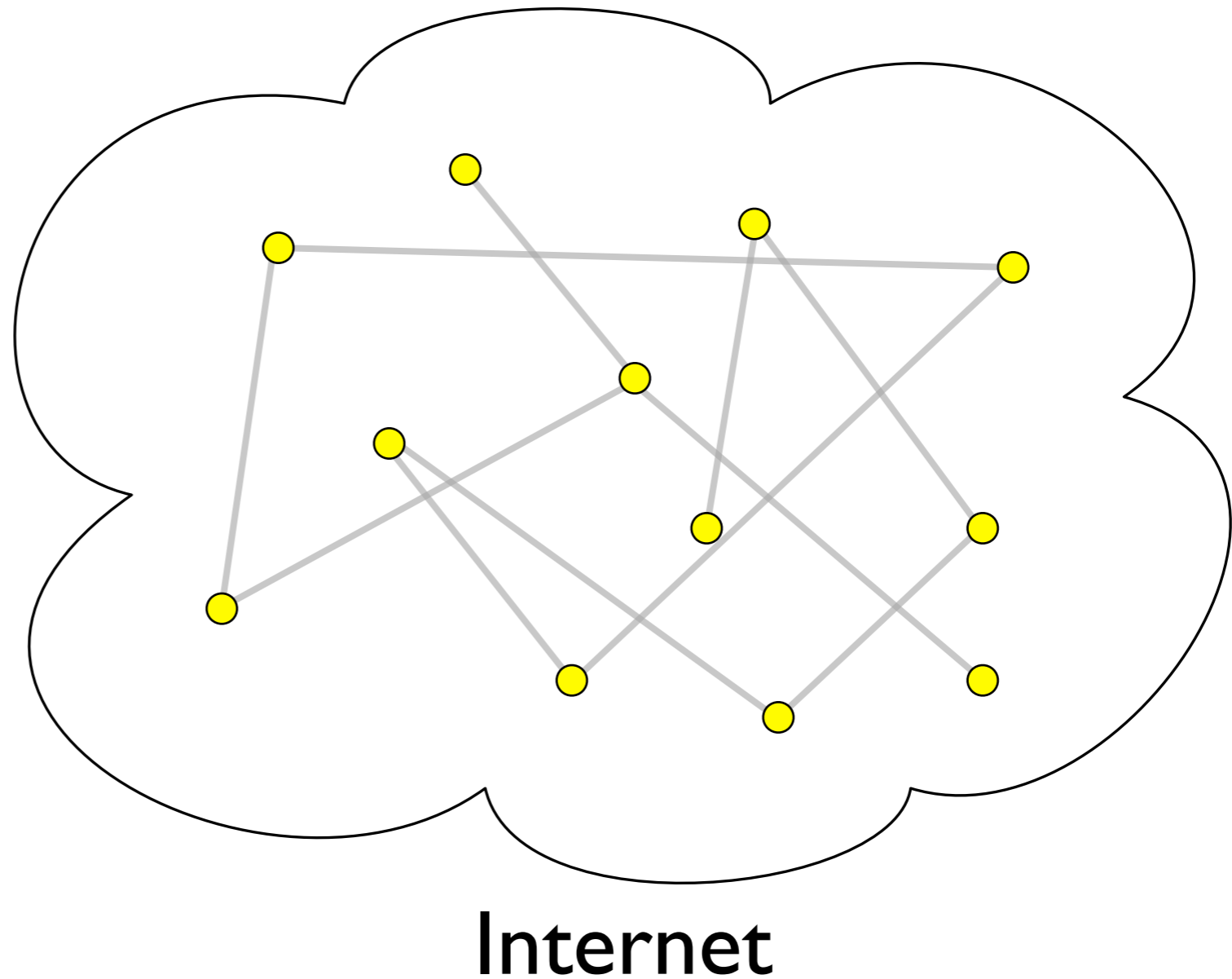
Several existing P2P applications

- **File sharing**
BitTorrent, Gnutella
- **Communication**
Skype
- **Caching**
Coral
- **Censorship-resistant publishing**
Freenet, Freehaven
- **Search engines**
YaCy, FAROO, Sixearch



Several existing P2P applications

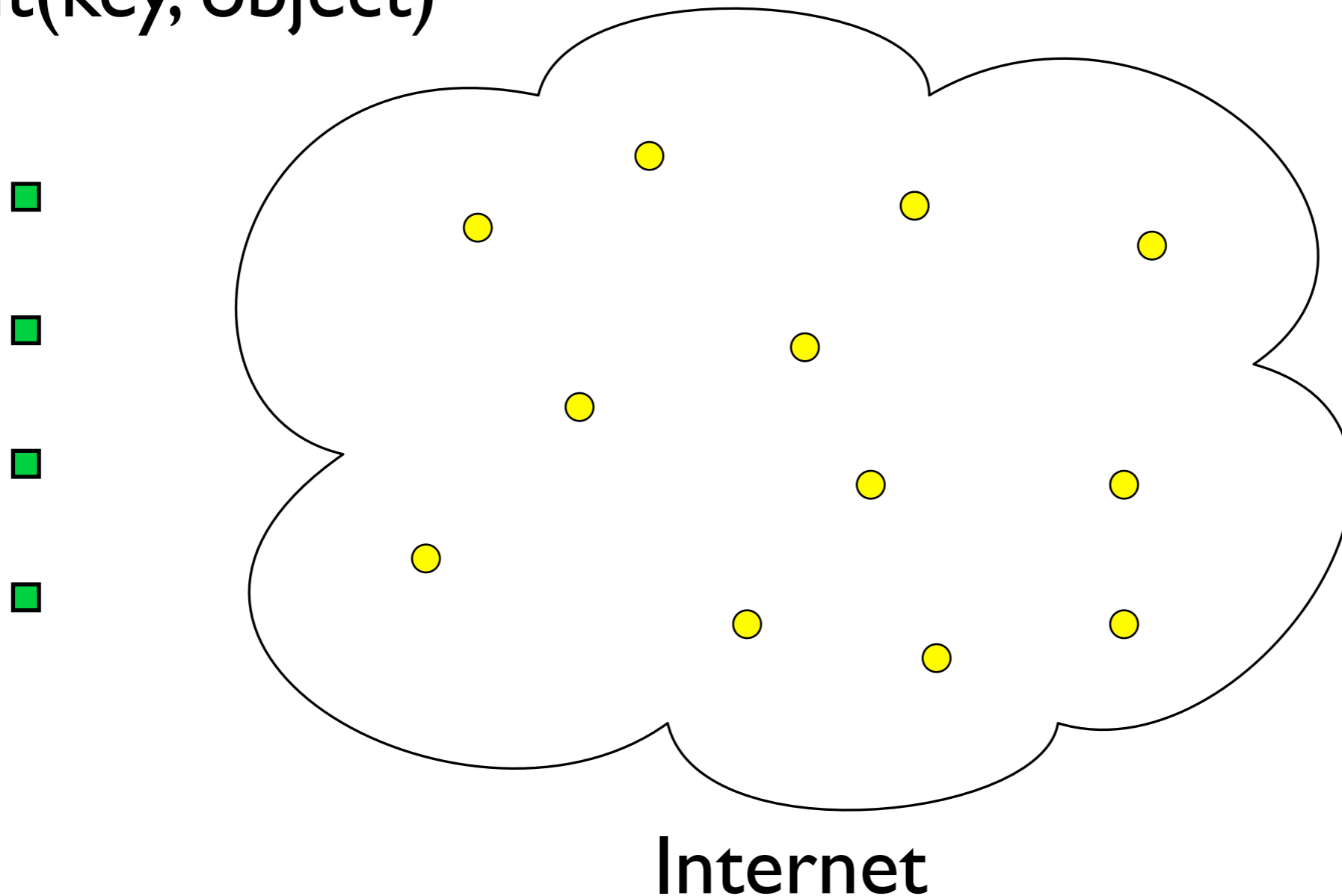
- **File sharing**
BitTorrent, Gnutella
- **Communication**
Skype
- **Caching**
Coral
- **Censorship-resistant publishing**
Freenet, Freehaven
- **Search engines**
YaCy, FAROO, Sixearch



How to distribute storage of data?

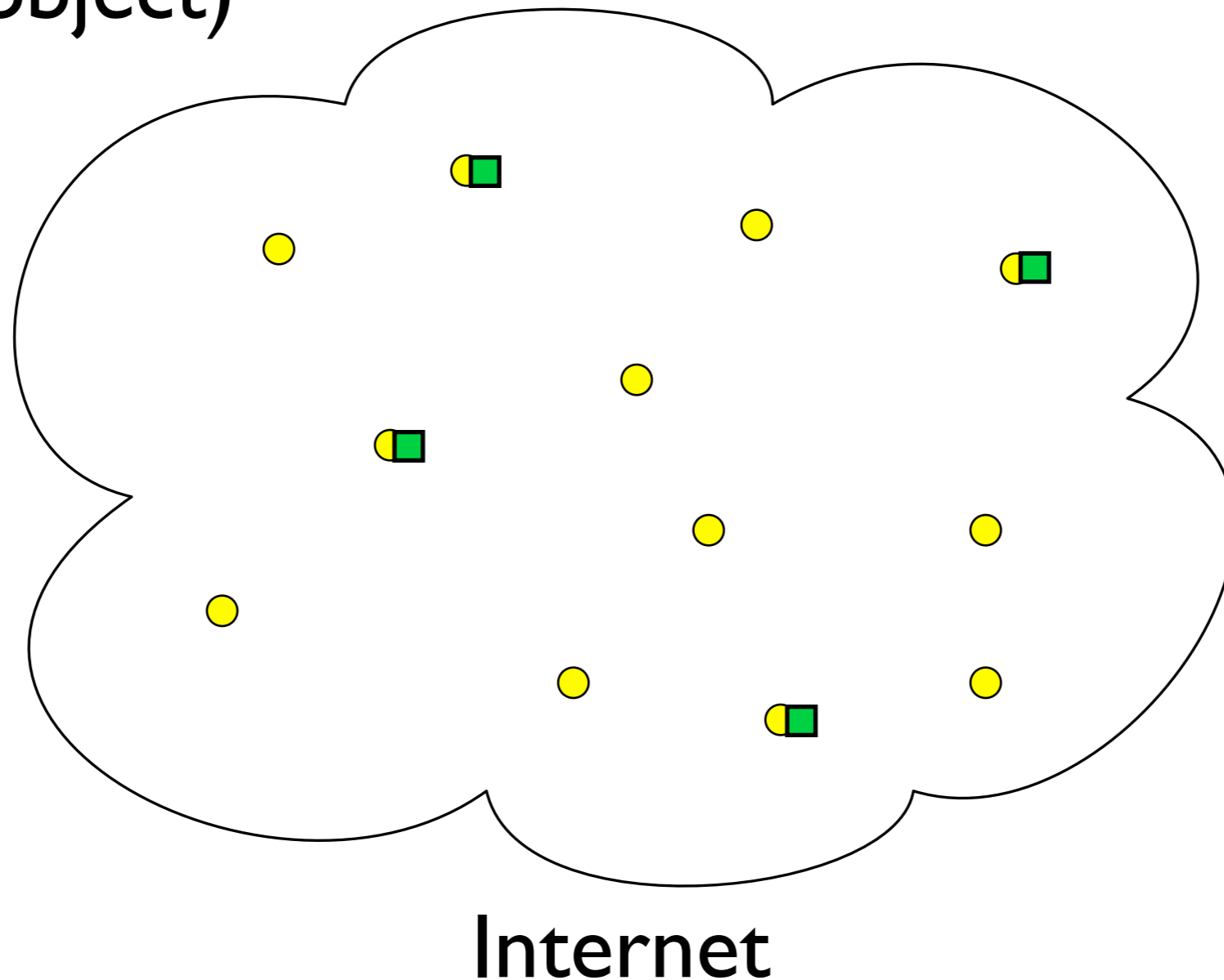
Distributed Hash Tables (DHTs) offer a distributed put-get interface

put(key, object)

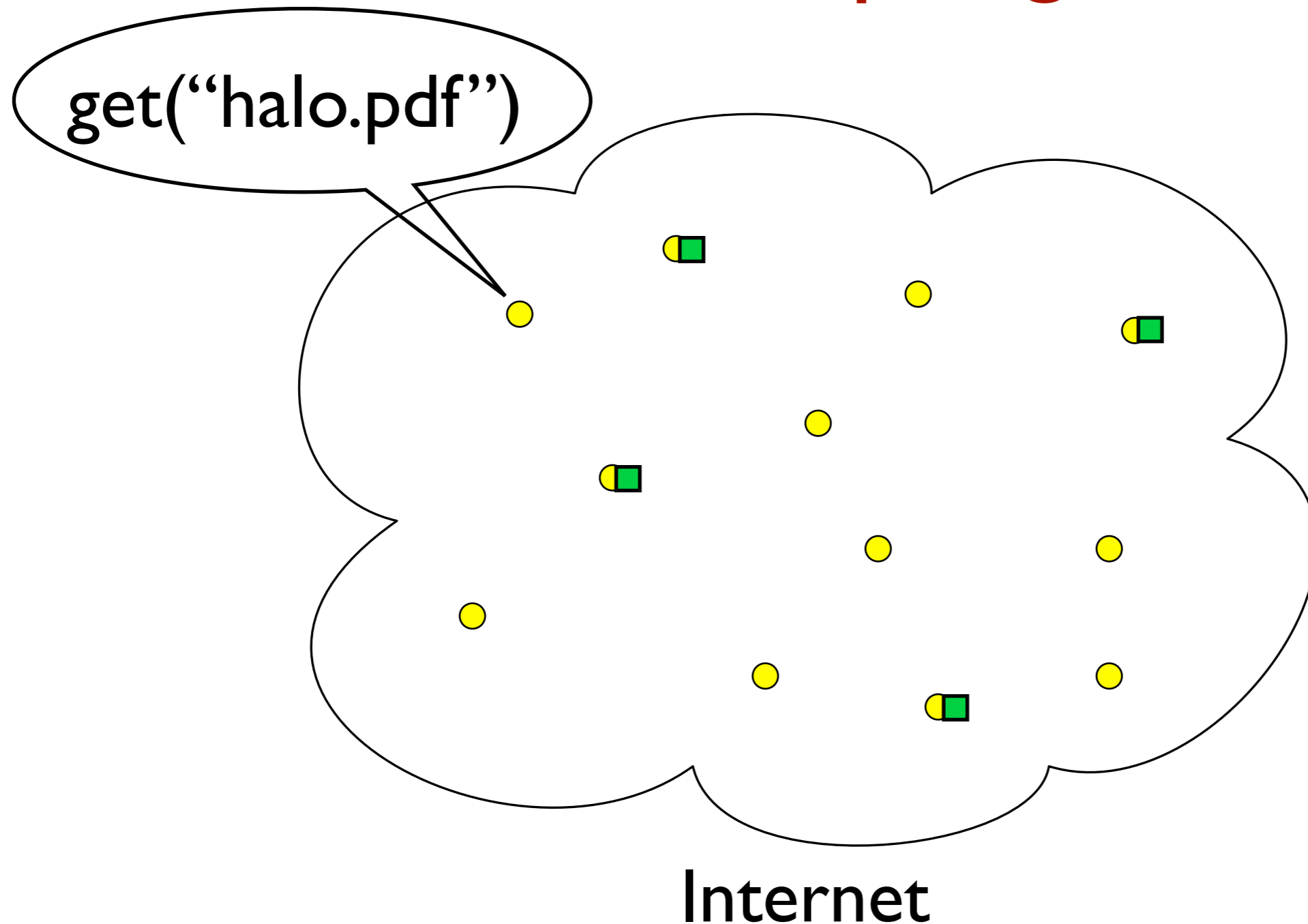


Distributed Hash Tables (DHTs) offer a distributed put-get interface

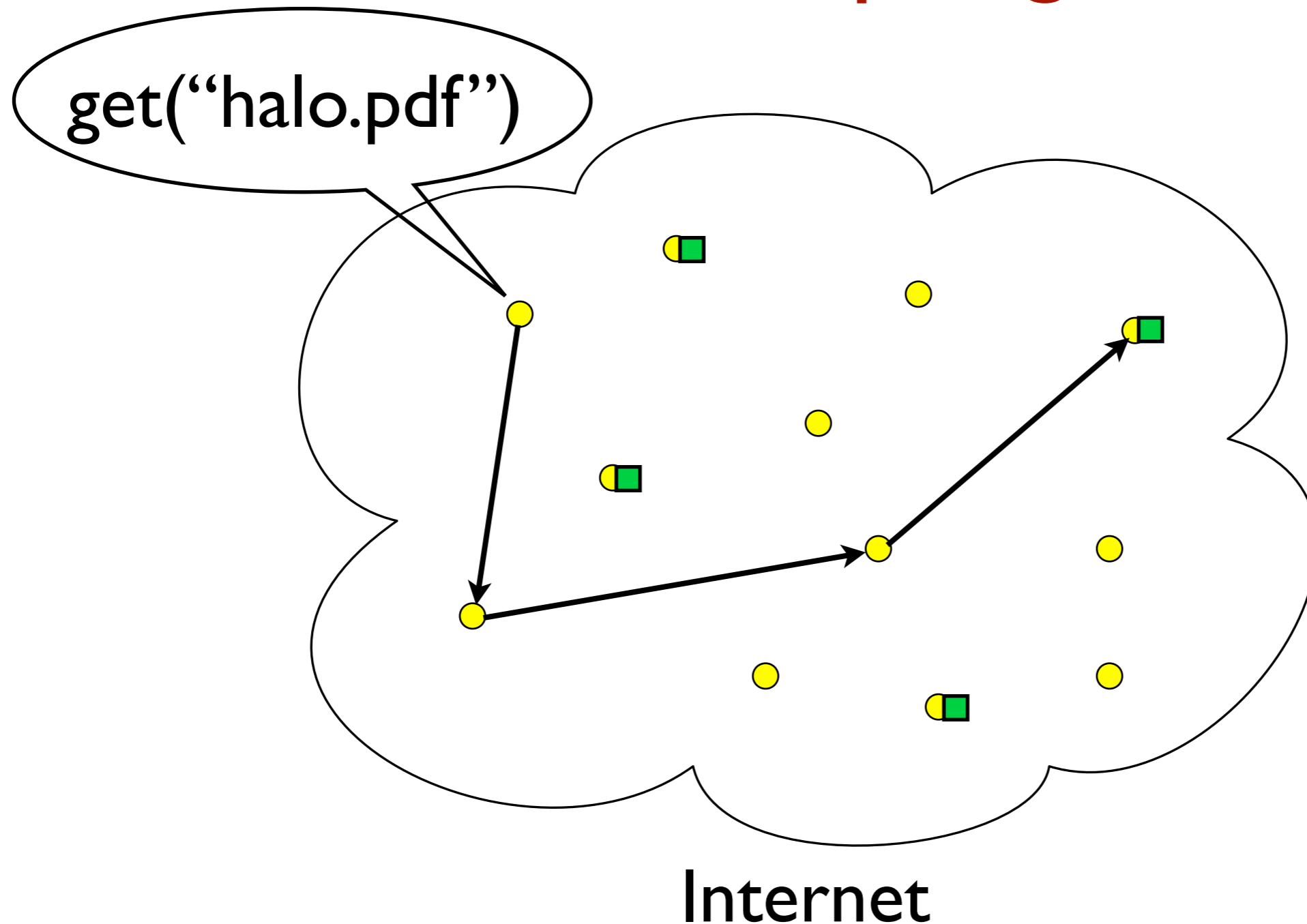
put(key, object)



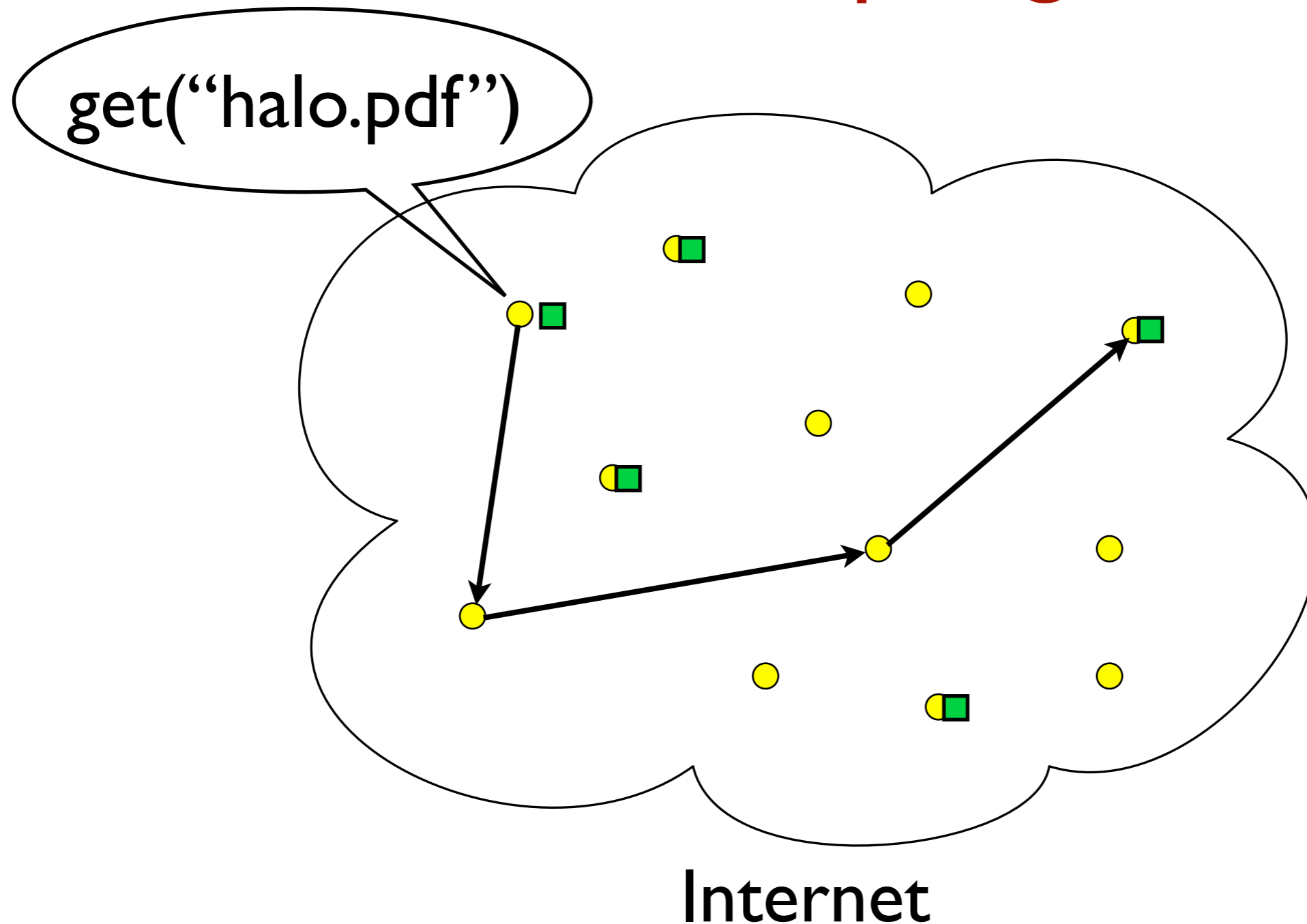
Distributed Hash Tables (DHTs) offer a distributed put-get interface



Distributed Hash Tables (DHTs) offer a distributed put-get interface



Distributed Hash Tables (DHTs) offer a distributed put-get interface



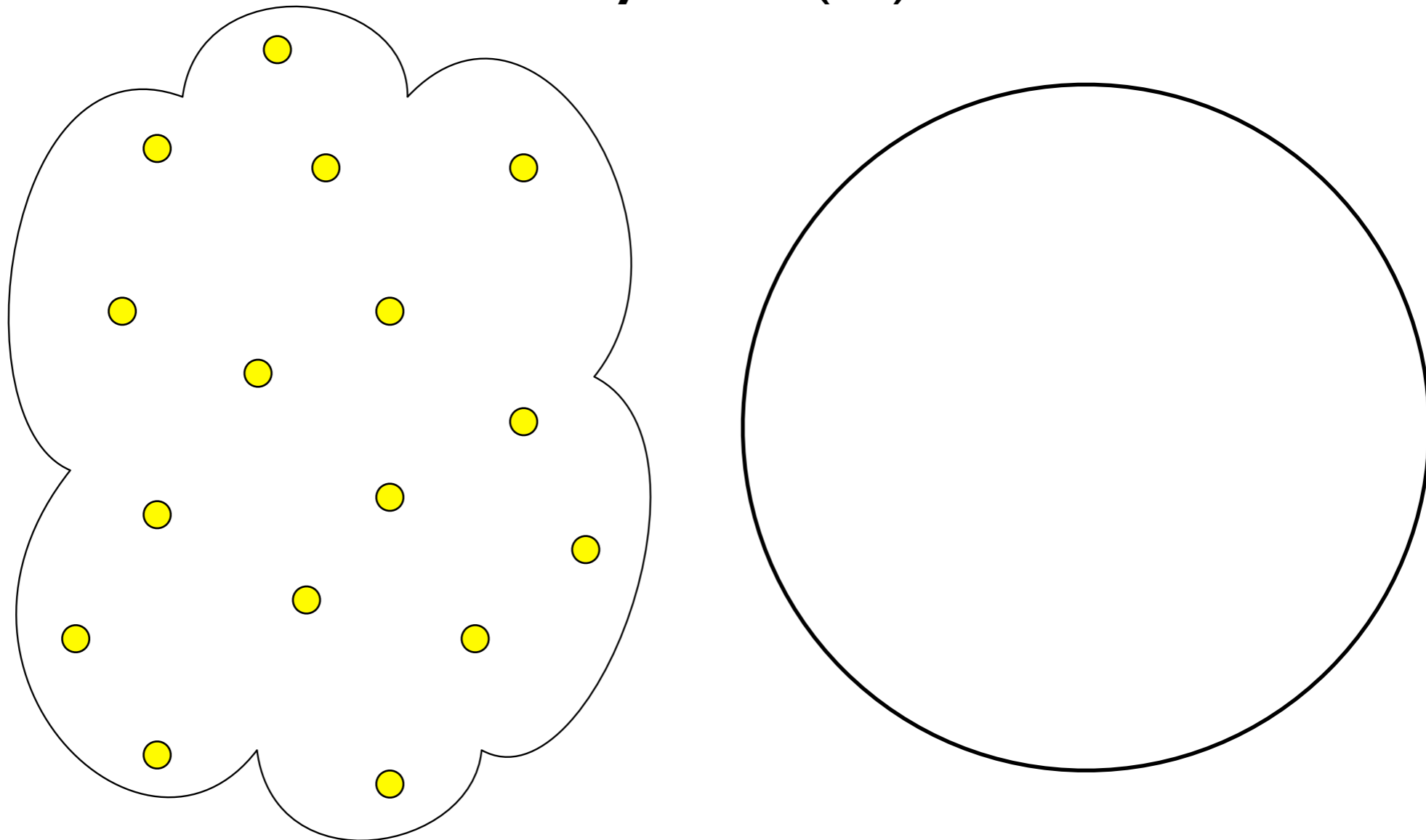
Chord

A well-studied DHT

Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan
SIGCOMM 2001

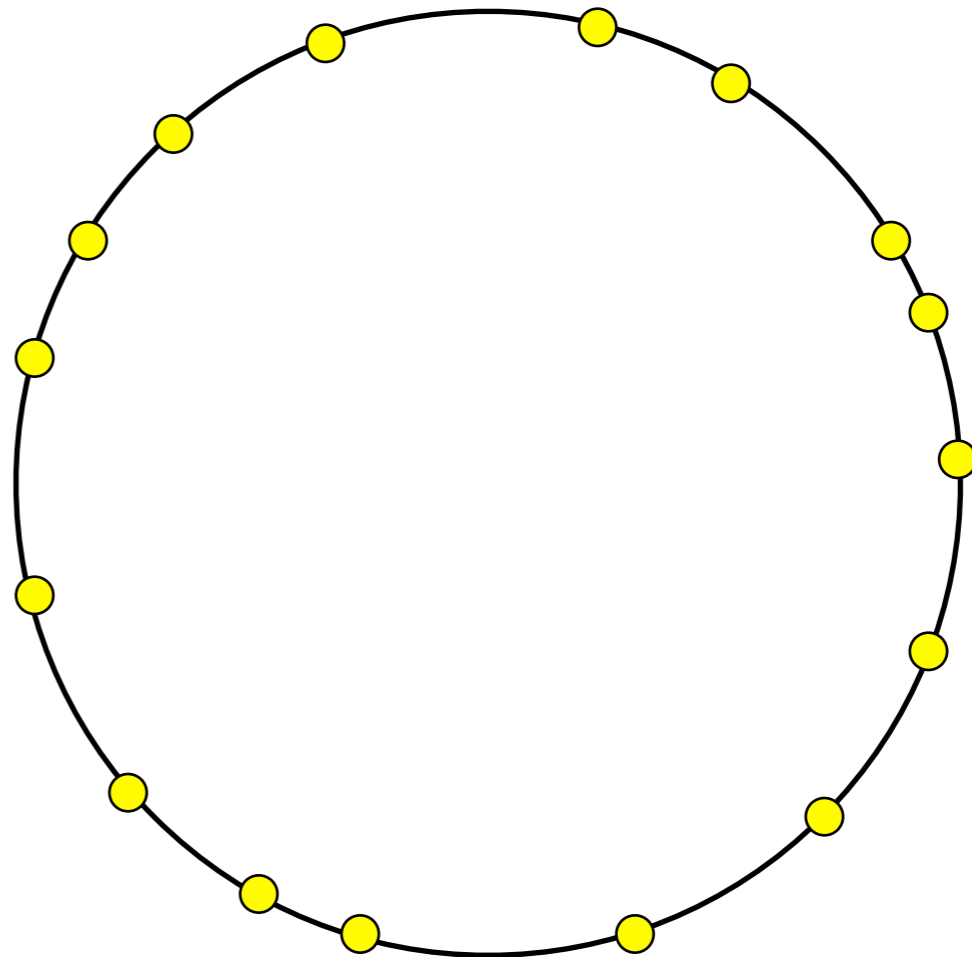
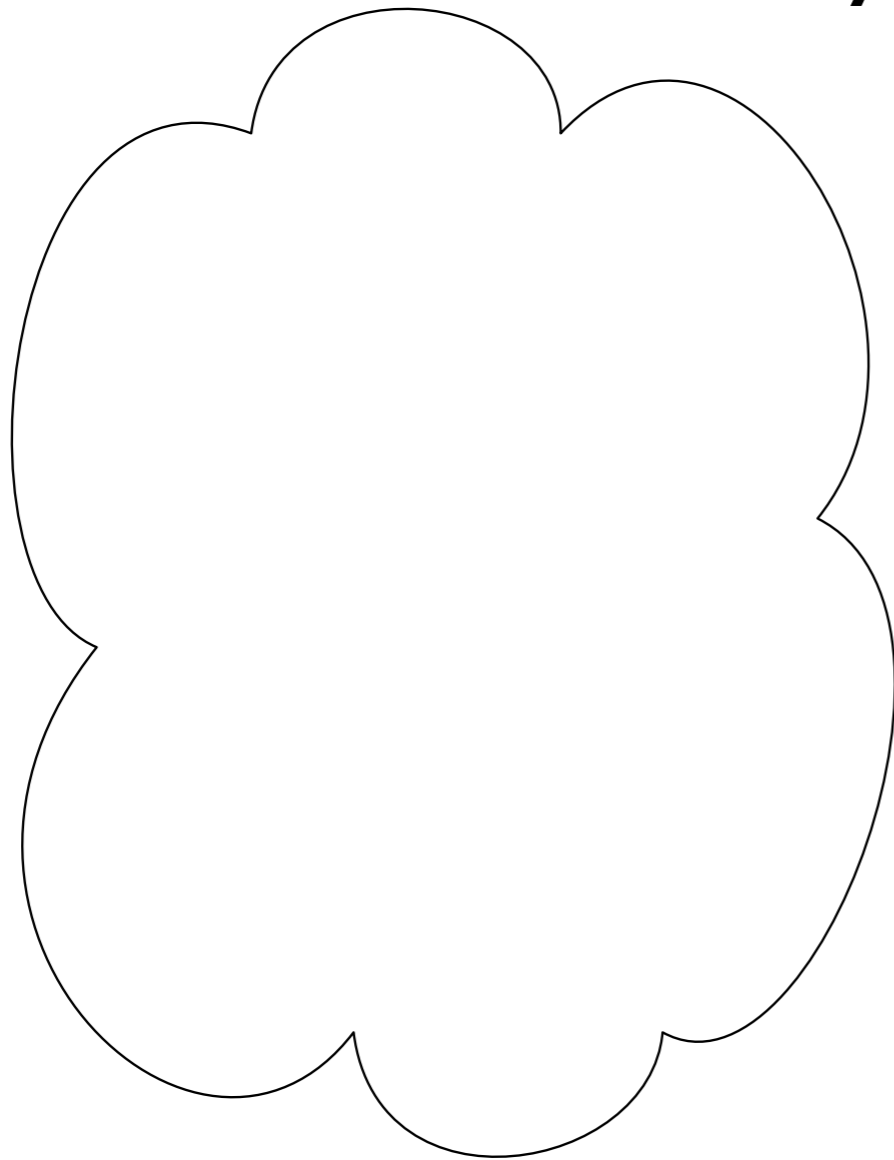
Chord maps nodes onto a ring

$$\text{key} = H(\text{IP})$$



Chord maps nodes onto a ring

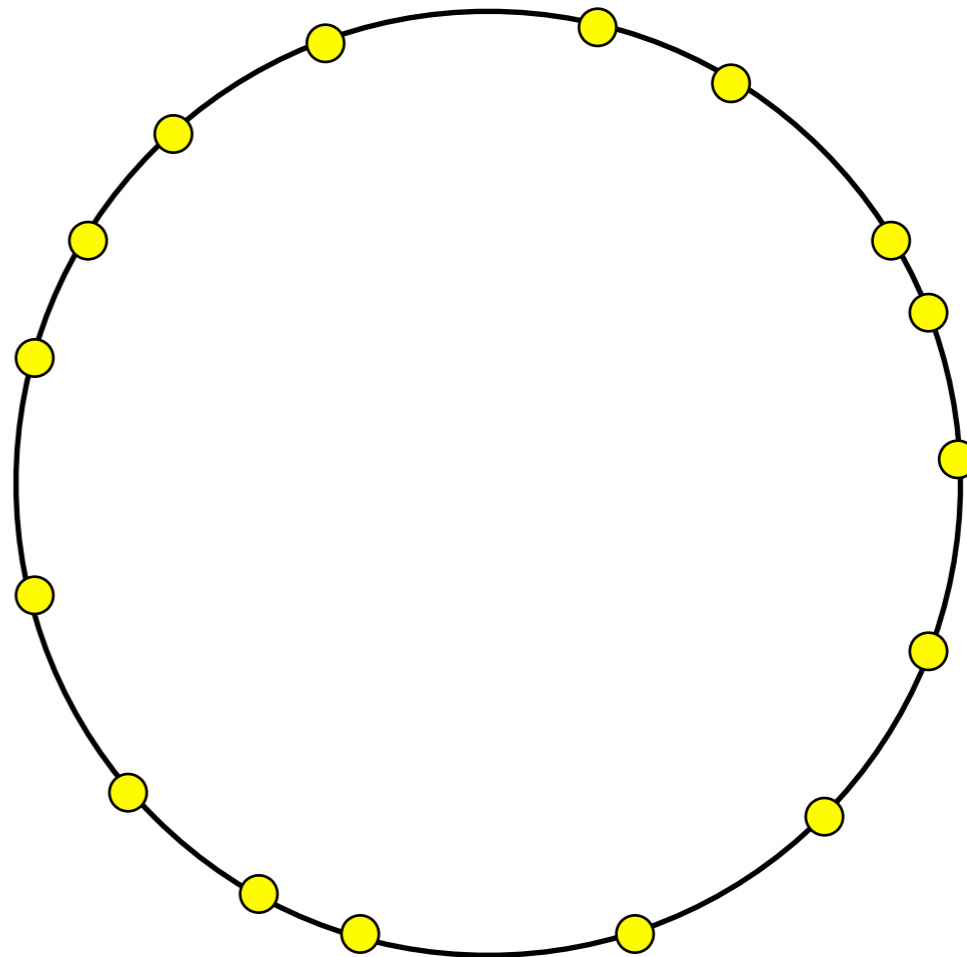
$$\text{key} = H(\text{IP})$$



put-get by mapping onto ring

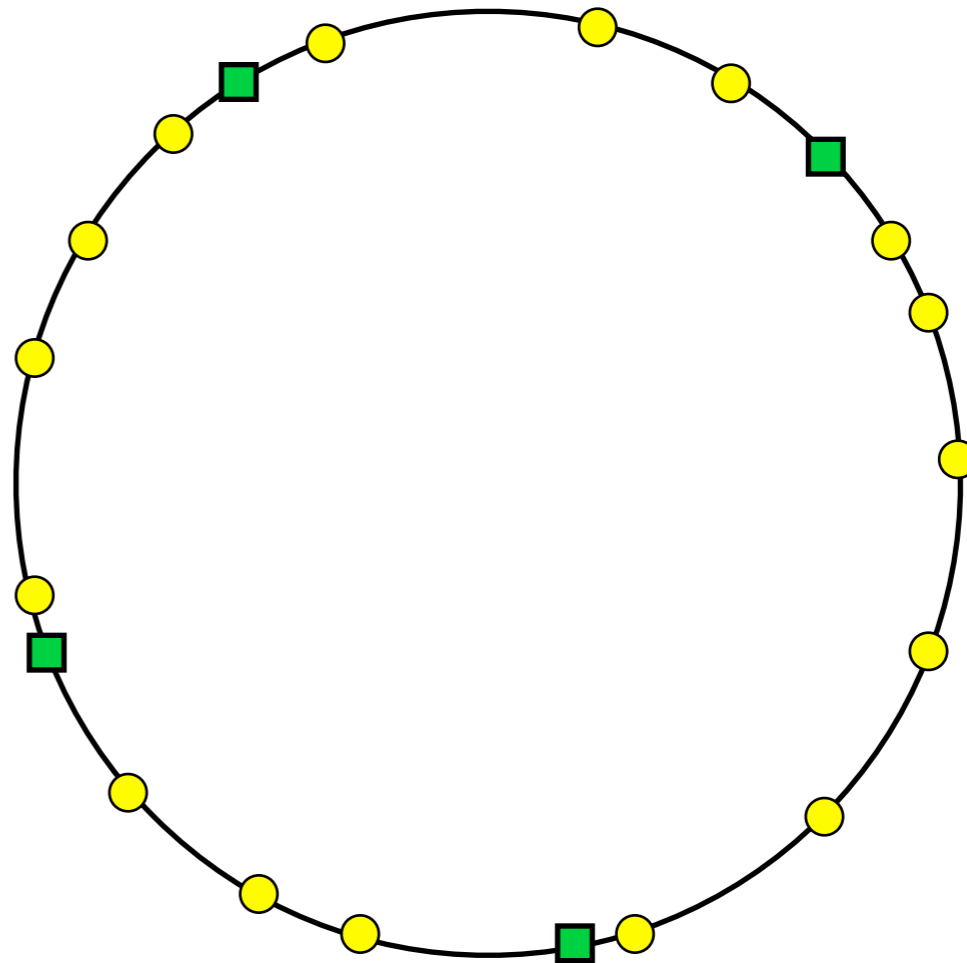
key = H(name)

- halo.pdf
- obama.jpg
- apple-fy08-report.pdf
- star-wars-trailer.mov



put-get by mapping onto ring

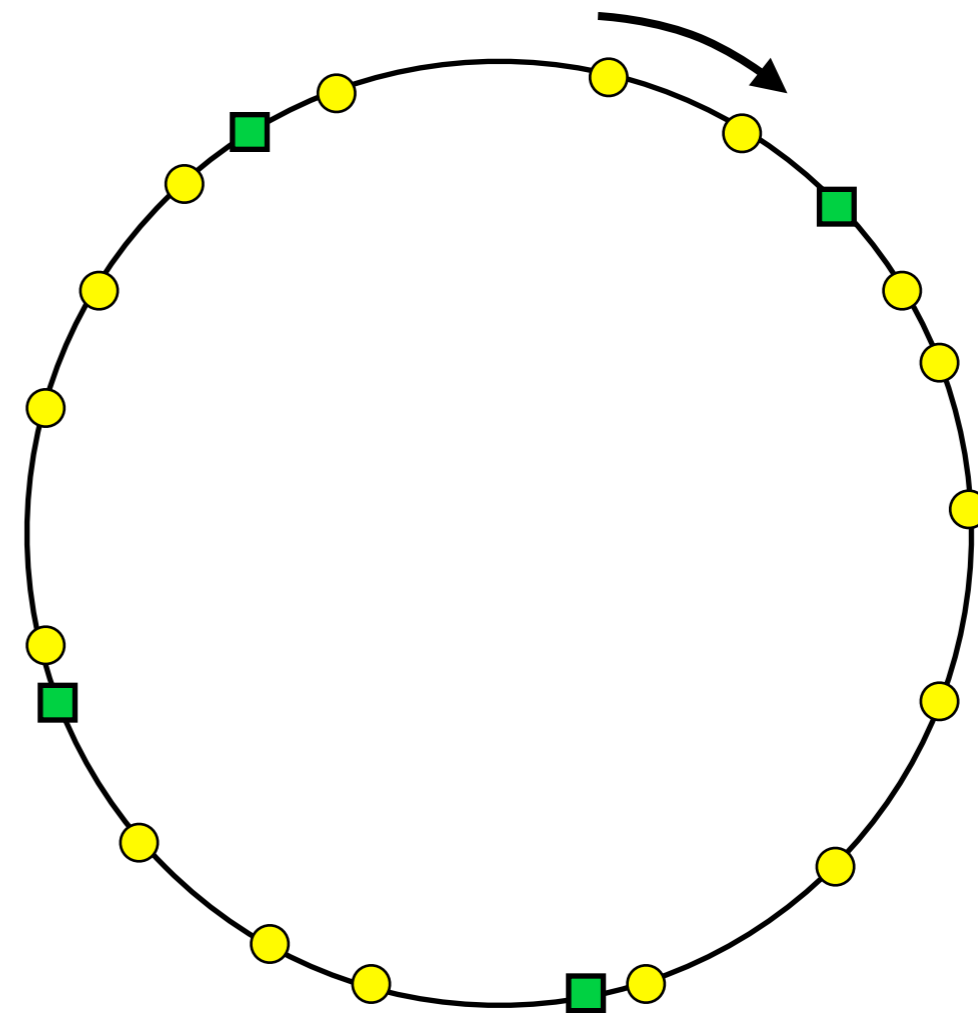
key = H(name)



put-get by mapping onto ring

$$\text{key} = H(\text{name})$$

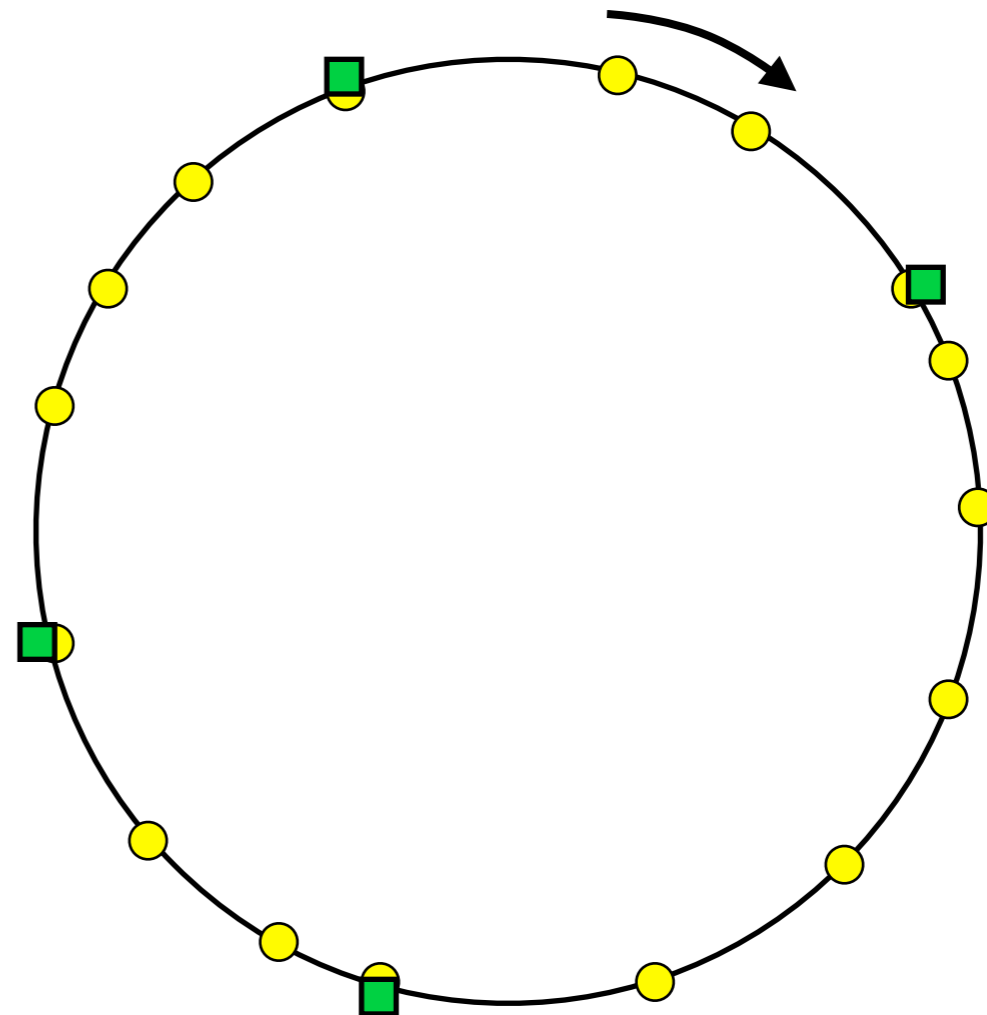
Resource “owned” by
clockwise-nearest node



put-get by mapping onto ring

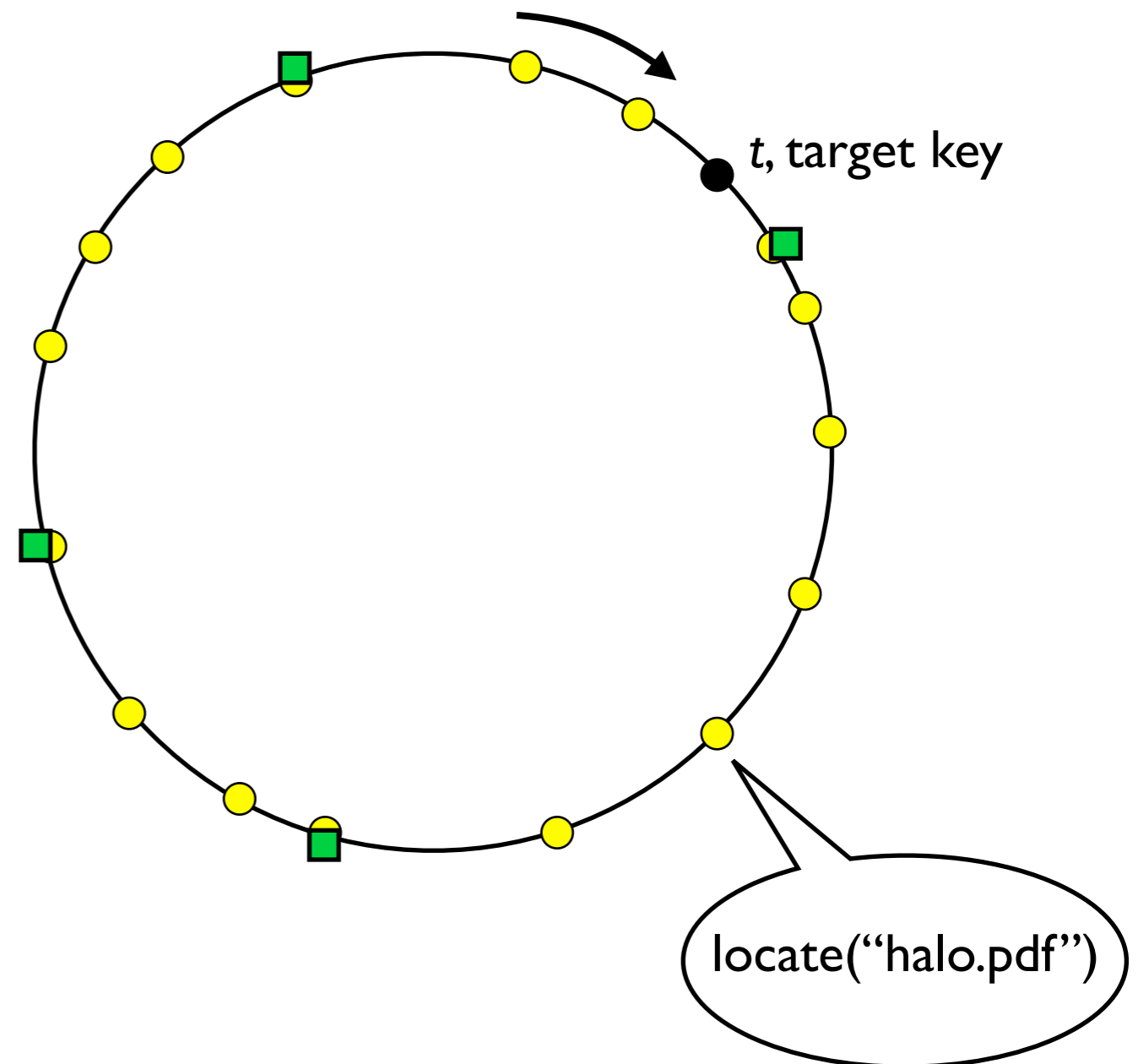
$$\text{key} = H(\text{name})$$

Resource “owned” by
clockwise-nearest node



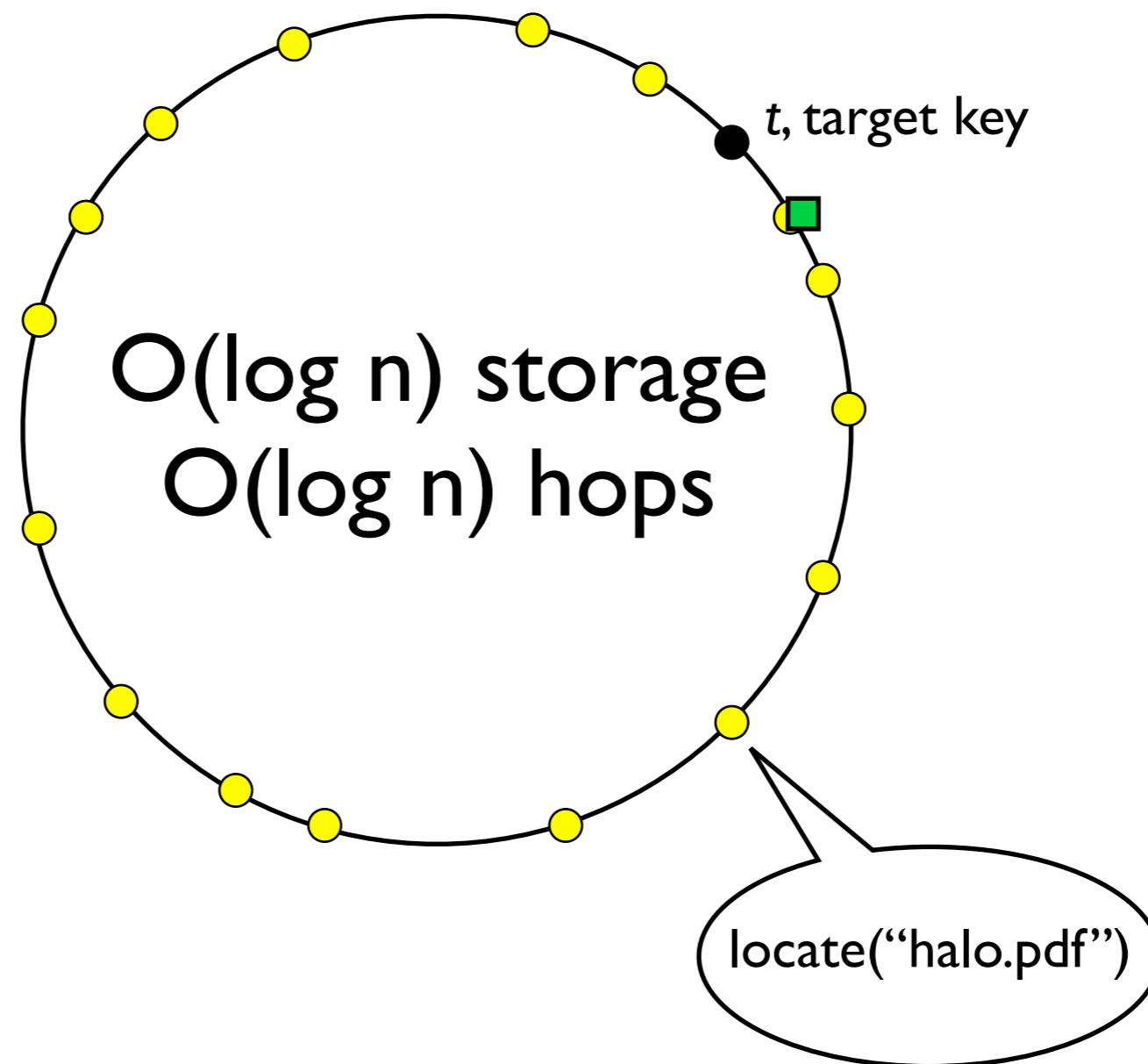
put-get by mapping onto ring

$$\text{key} = H(\text{name})$$

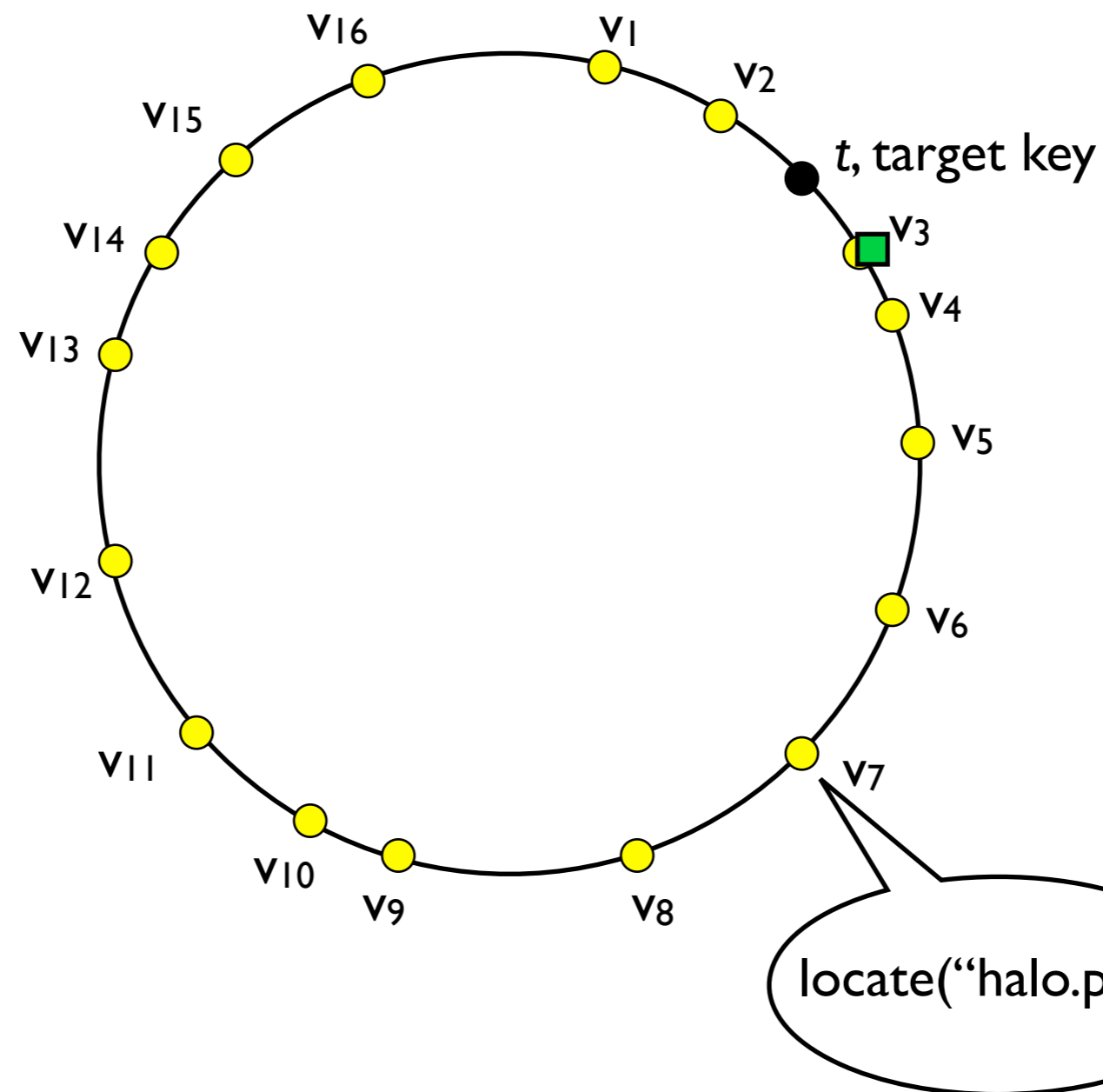


Next question:
How to *locate* owners?
`locate("halo.pdf")`

Route using “fingers”



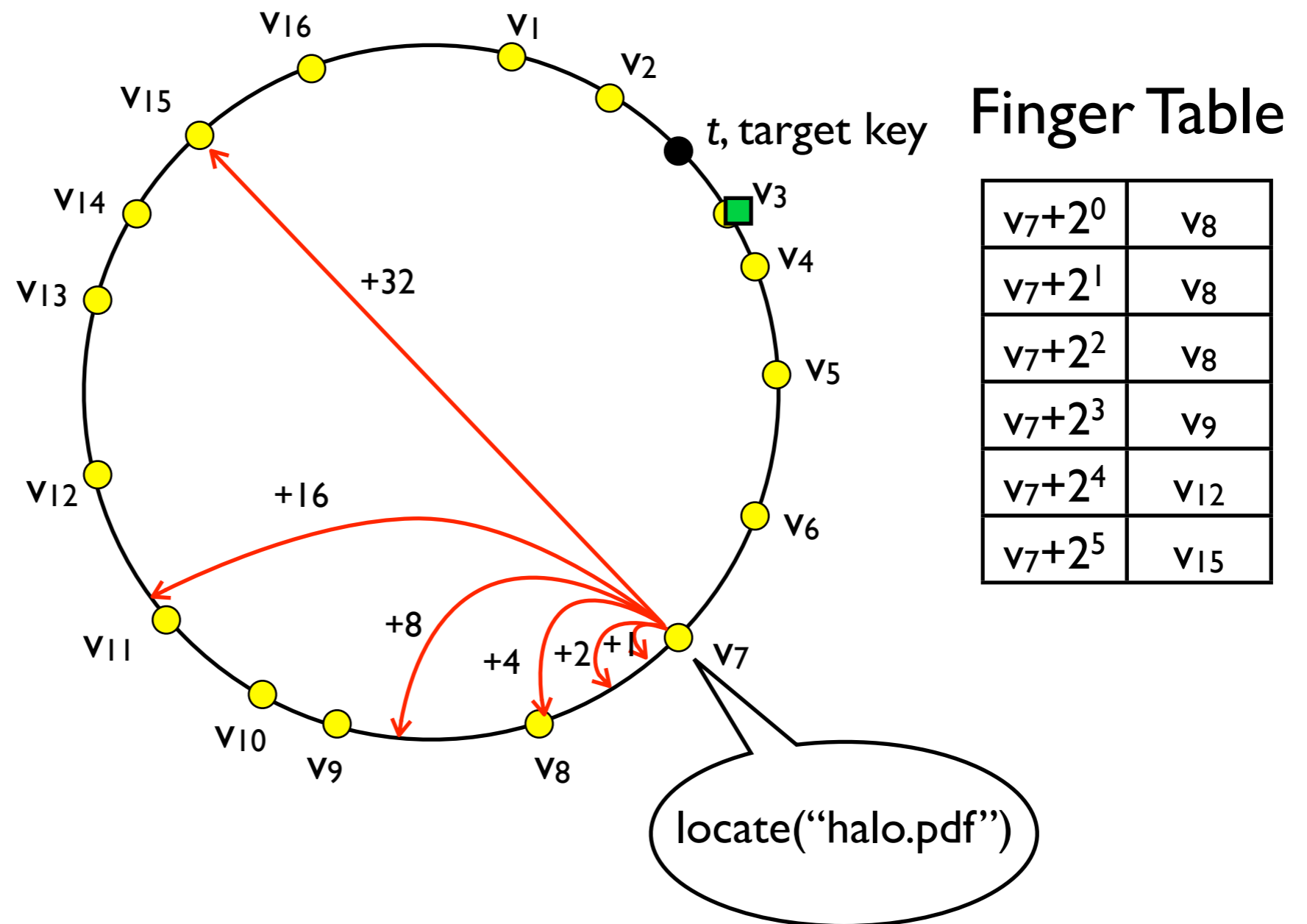
Route using “fingers”



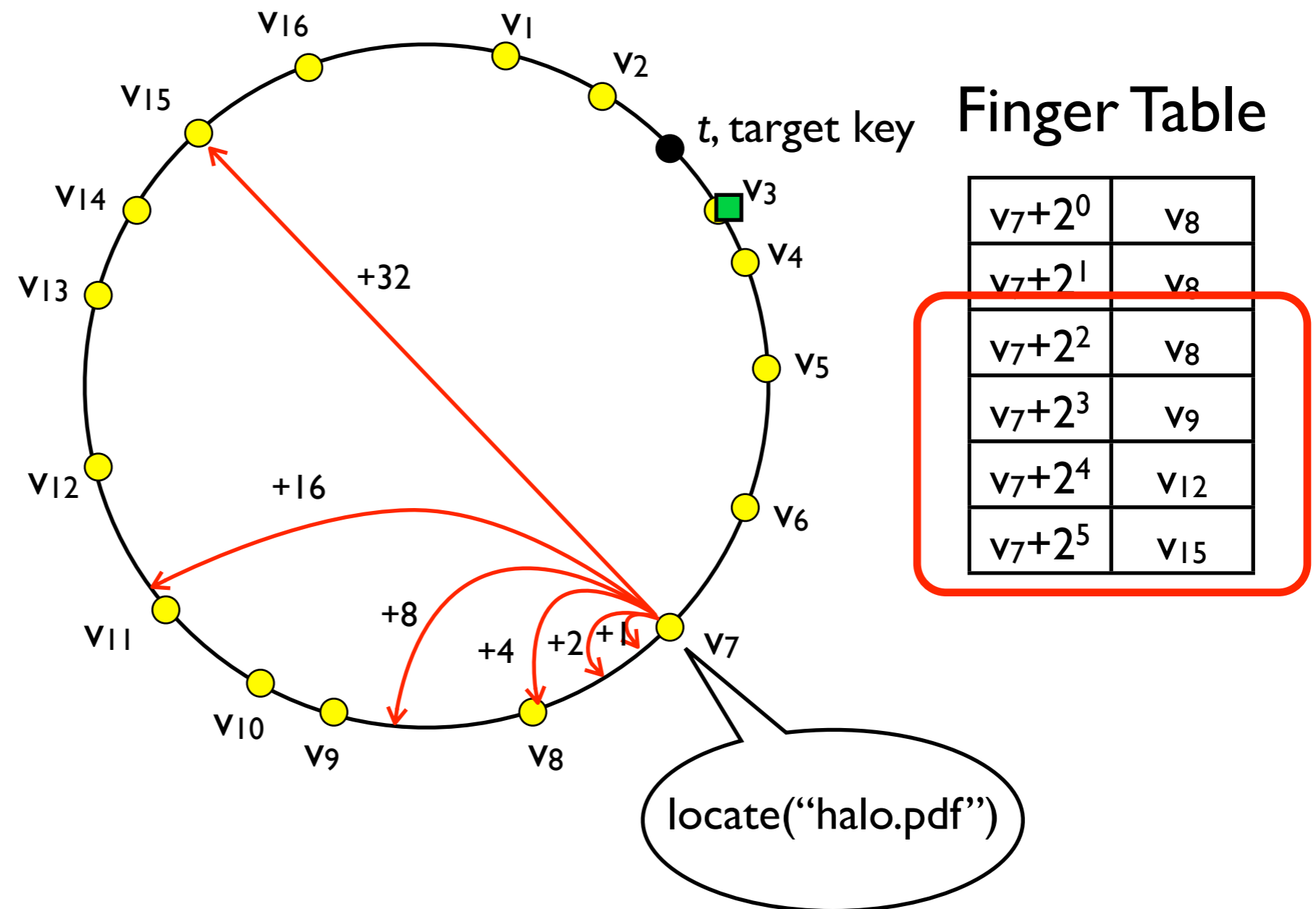
Finger Table

v_7+2^0	v_8
v_7+2^1	v_8
v_7+2^2	v_8
v_7+2^3	v_9
v_7+2^4	v_{12}
v_7+2^5	v_{15}

Route using “fingers”

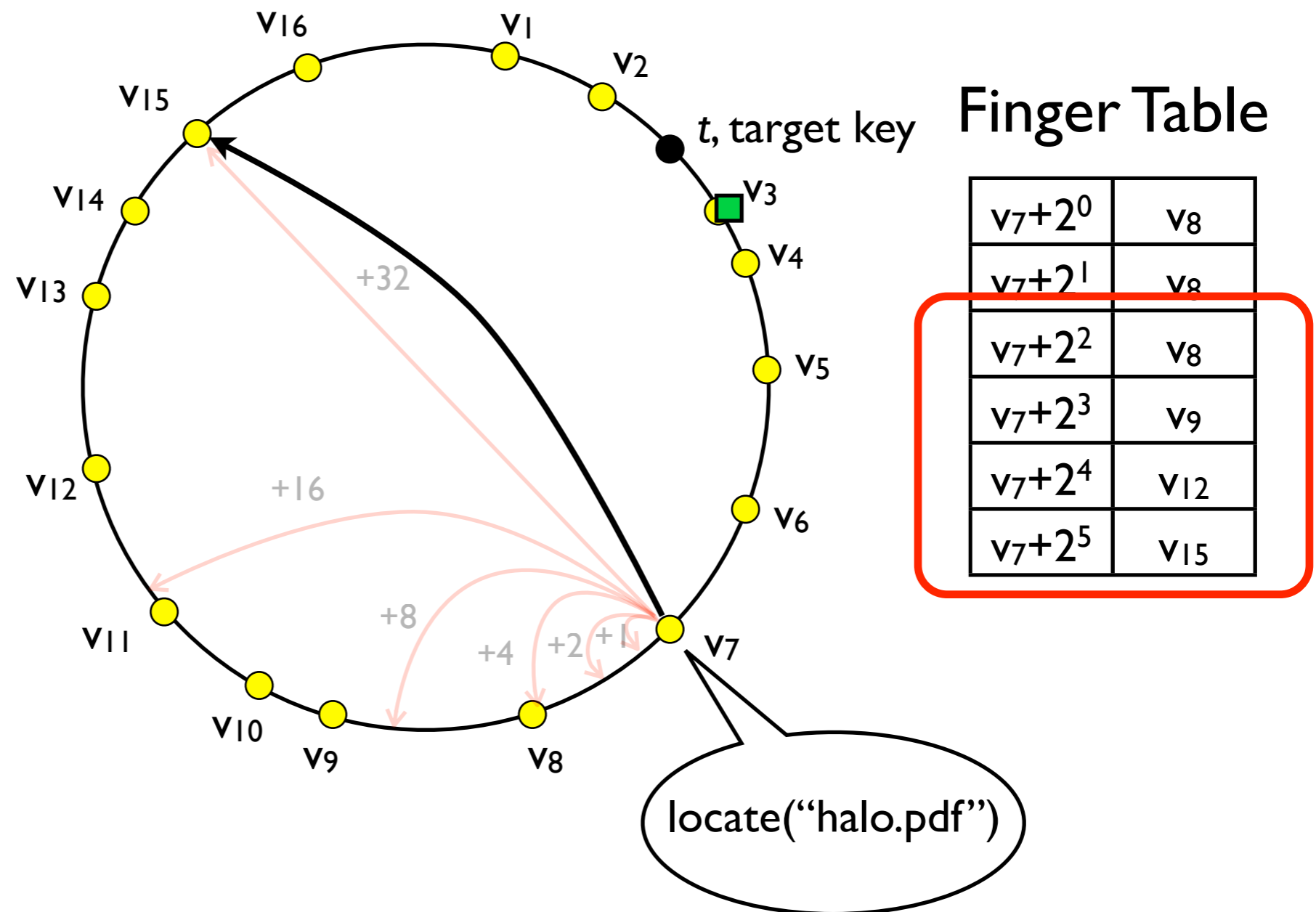


Route using “fingers”



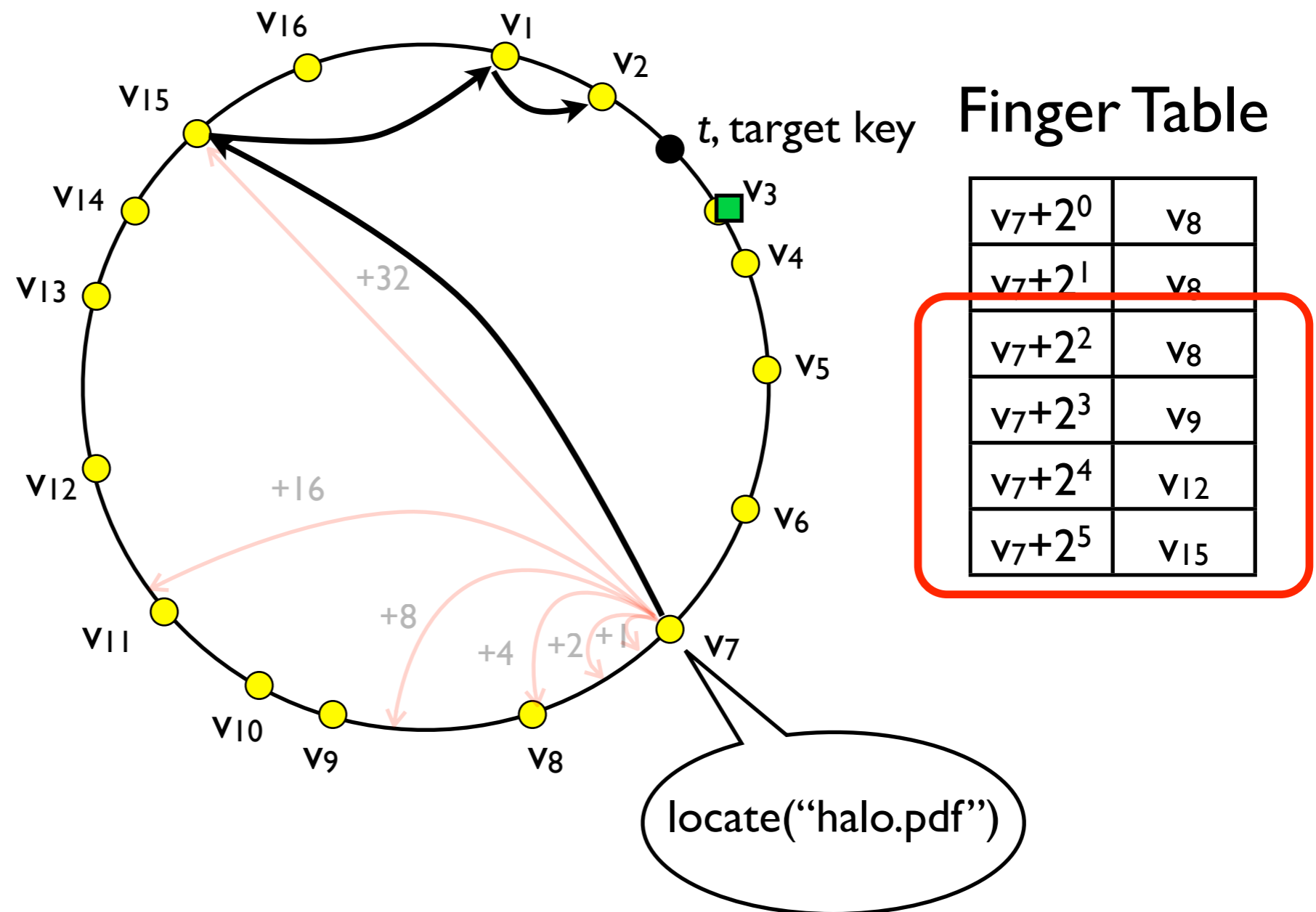
Route using “fingers”

Observation: Can always make it half way to any key



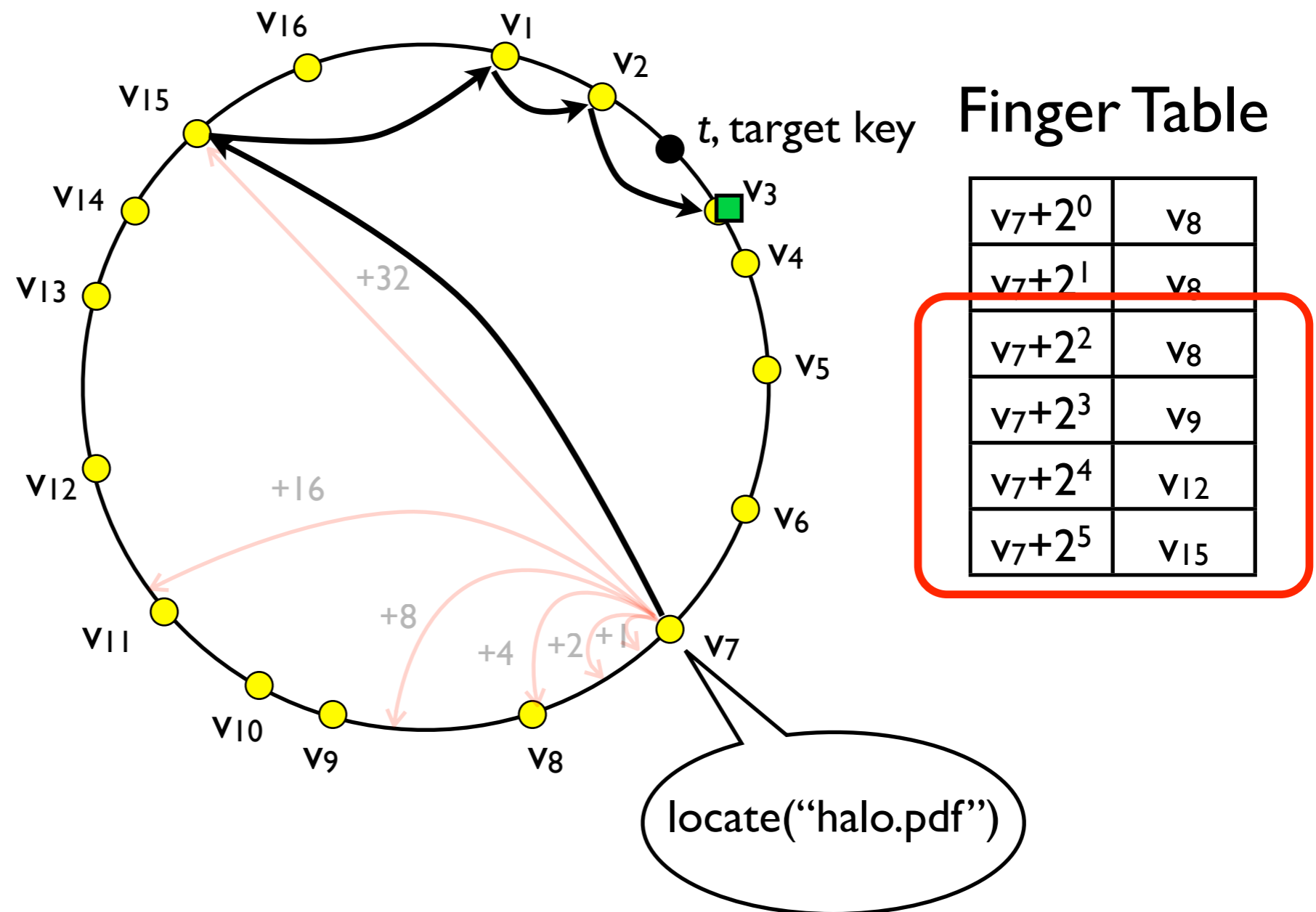
Route using “fingers”

Observation: Can always make it half way to any key



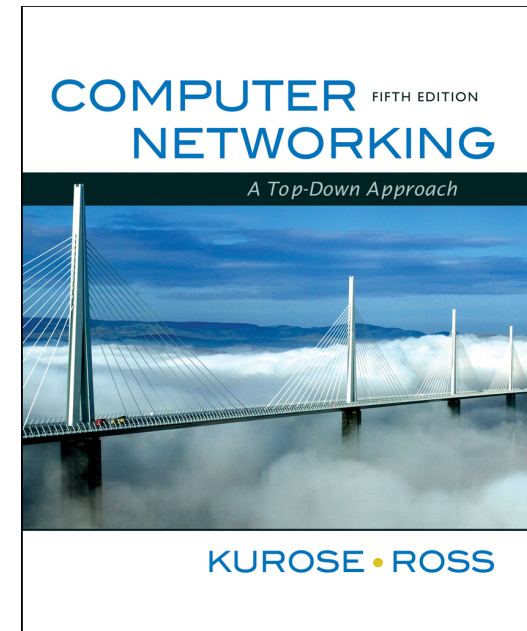
Route using “fingers”

Observation: Can always make it half way to any key



Chapter 2

Application Layer



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- ❖ If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- ❖ If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2010
J.F Kurose and K.W. Ross, All Rights Reserved

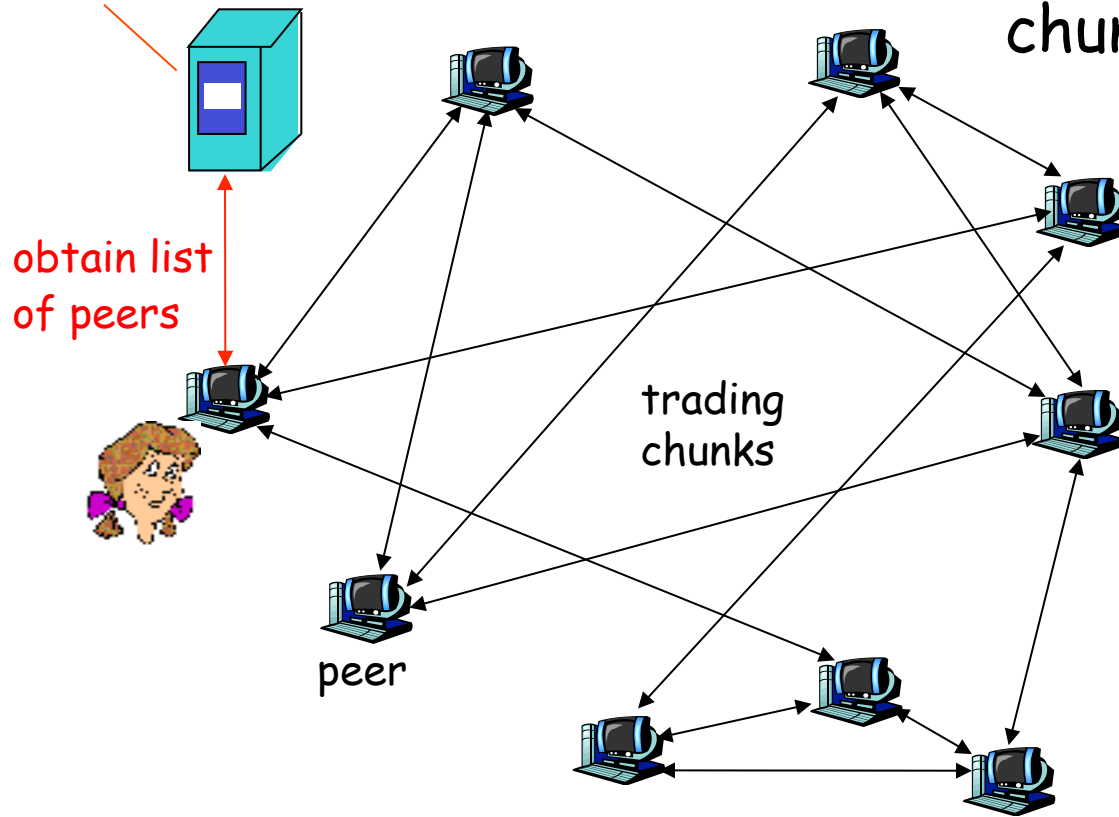
*Computer Networking:
A Top Down Approach,
5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April
2009.*

File distribution: BitTorrent

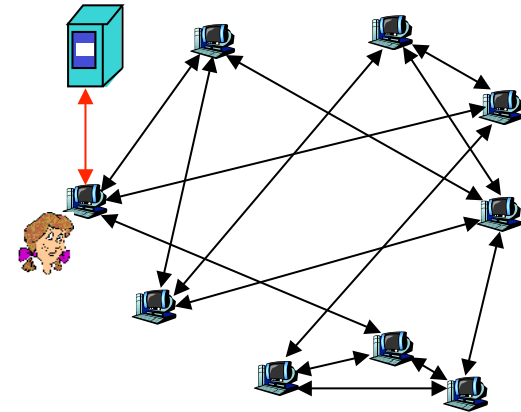
P2P file distribution

tracker: tracks peers participating in torrent

torrent: group of peers exchanging chunks of a file



BitTorrent (1)



- ❖ file divided into 256KB *chunks*.
- ❖ peer joining torrent:
 - has no chunks, but will accumulate them over time
 - registers with tracker to get list of peers, connects to subset of peers (“neighbors”)
- ❖ while downloading, peer uploads chunks to other peers.
- ❖ peers may come and go
- ❖ once peer has entire file, it may (selfishly) leave or (altruistically) remain

BitTorrent (2)

Pulling Chunks

- ❖ at any given time, different peers have different subsets of file chunks
- ❖ periodically, a peer (Alice) asks each neighbor for list of chunks that they have.
- ❖ Alice sends requests for her missing chunks
 - rarest first

Sending Chunks: tit-for-tat

- ❖ Alice sends chunks to four neighbors currently sending her chunks *at the highest rate*
 - re-evaluate top 4 every 10 secs
- ❖ every 30 secs: randomly select another peer, starts sending chunks
 - newly chosen peer may join top 4
 - “optimistically unchoke”

BitTorrent: Tit-for-tat

- (1) Alice “optimistically unchokes” Bob
- (2) Alice becomes one of Bob’s top-four providers; Bob reciprocates
- (3) Bob becomes one of Alice’s top-four providers

