Web Caching Overview

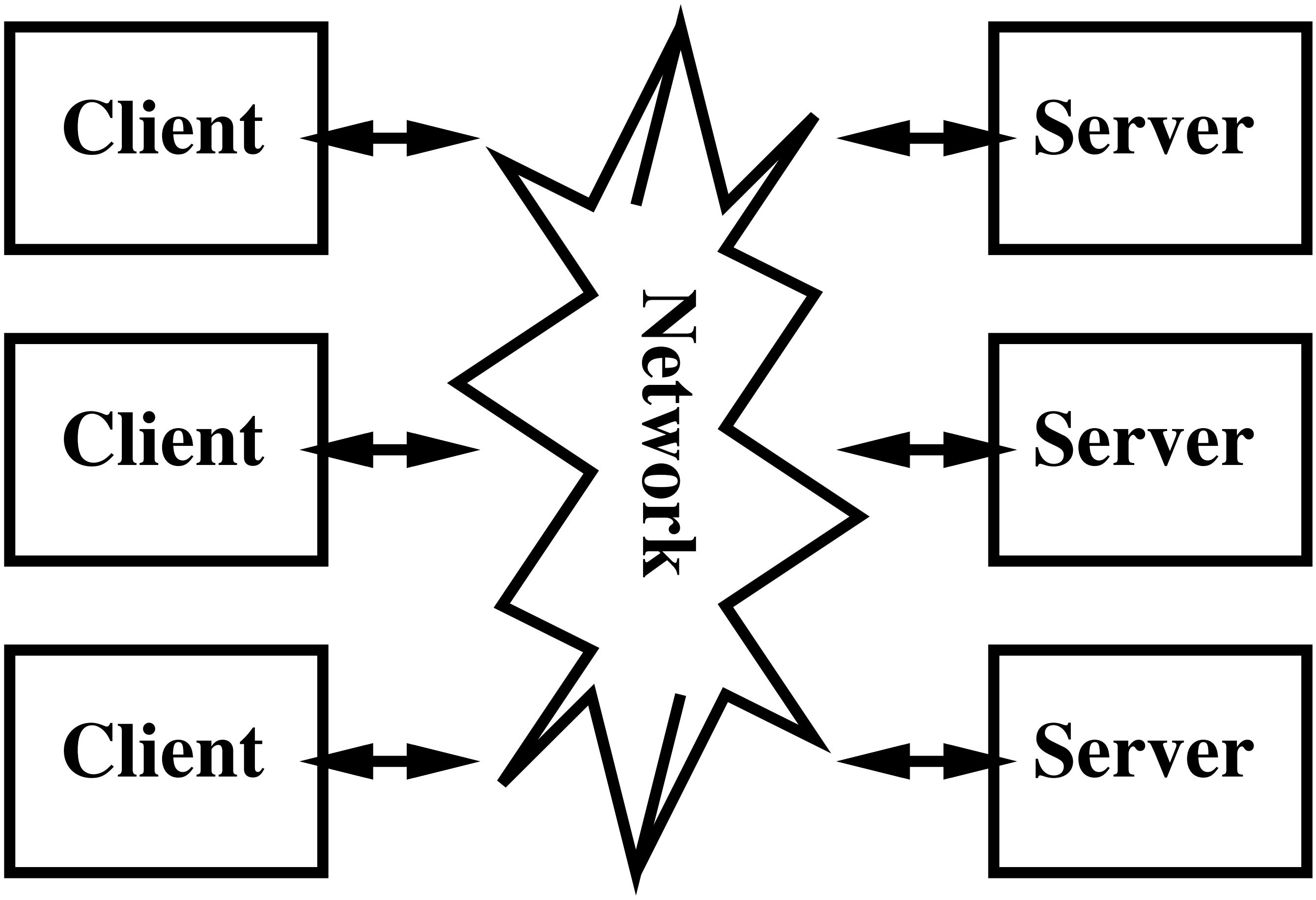
Vivek S. Pai

April 24, 2000

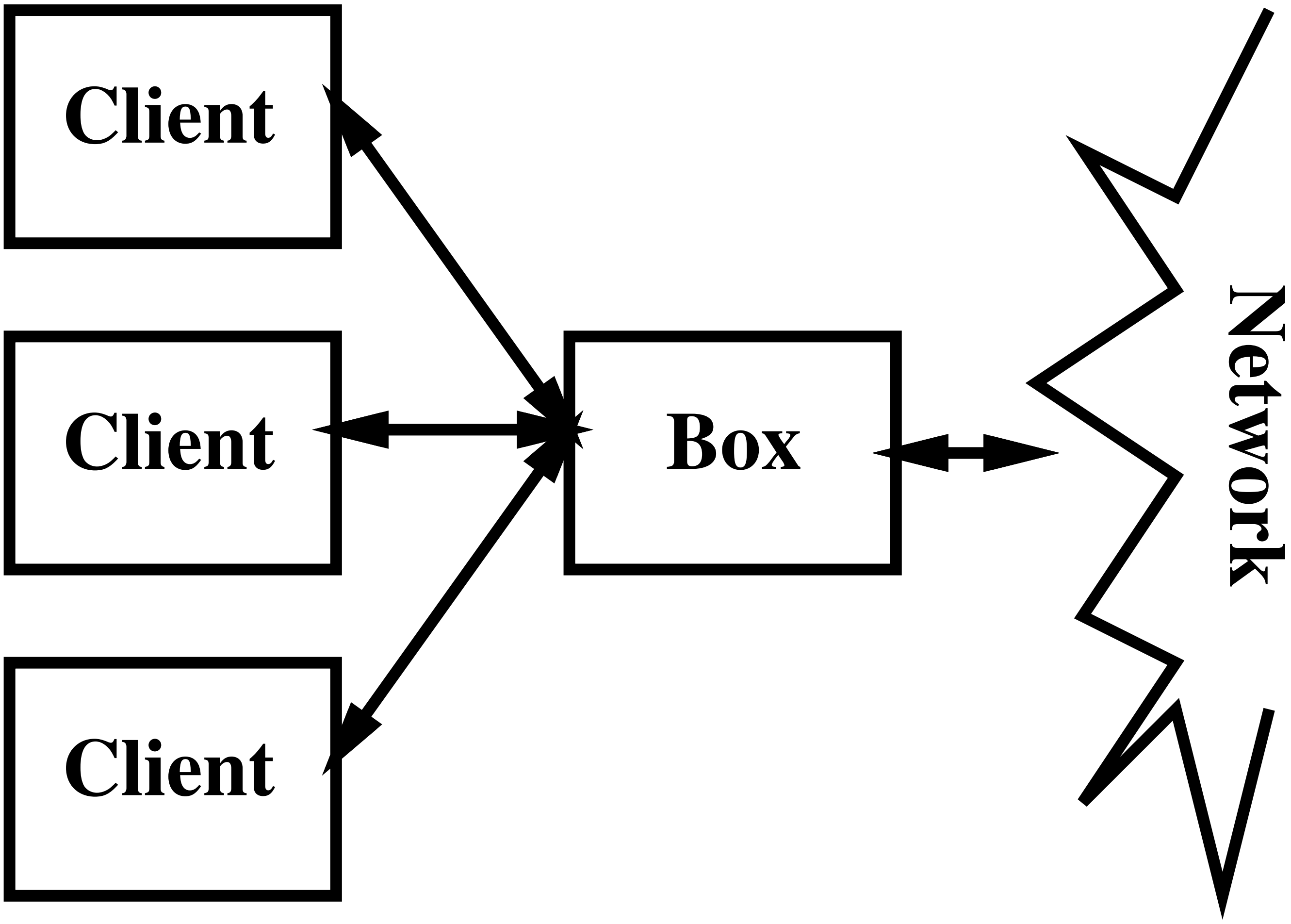
In The Beginning...

A Brief (and fake) History of Time

More Machines Appear



Machines Get Cheaper



Why Those Boxes Exist

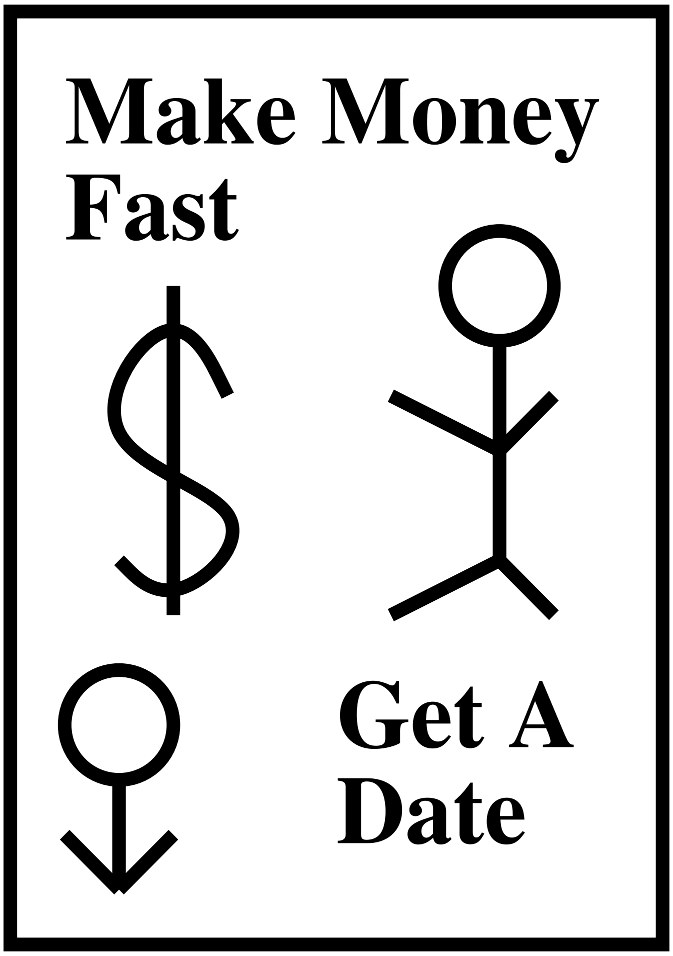
Main reasons

- routing
- security (firewalls)

Implications

- modified applications
- centralization

The Web Appears

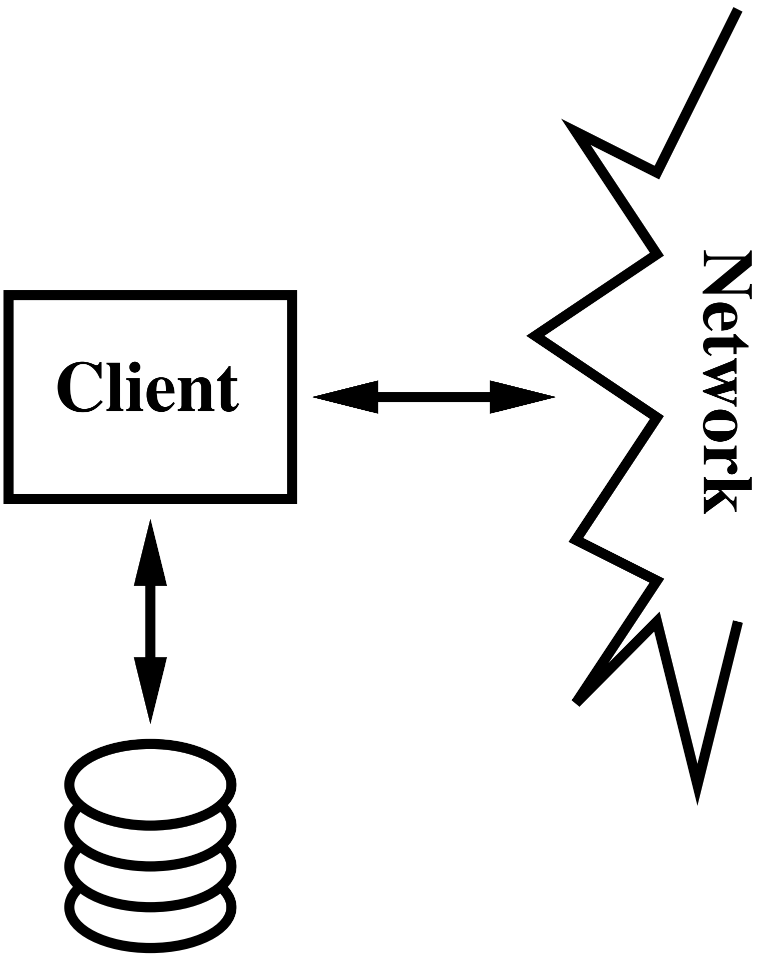


URLs:
http://www.somesite.com/.../main.html
http://www.somesite.com/.../picture.gif
http://www.somesite.com/.../money.jpg

People Talk to Each Other

- common interests appear
- sites become popular
- traffic not homogenous
- the web slows down

Browser Caches



1. check disk
2. possibly revalidate
3. fetch if needed

Cache Benefits

Reduces

- # of requests
- latency
- bandwidth

Problems

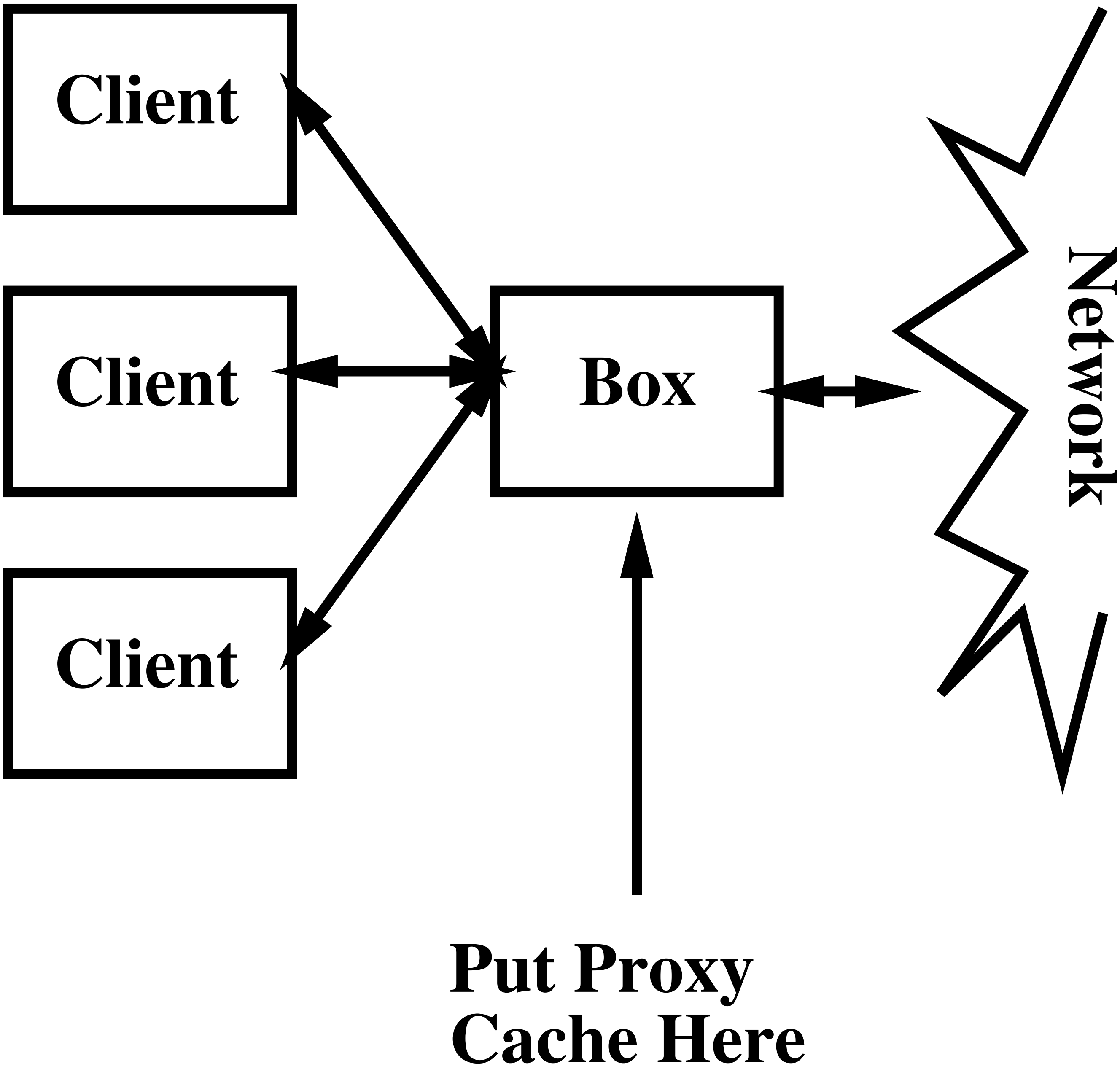
Types of misses

- compulsory
- capacity
- conflict

Compulsory - how many, really?

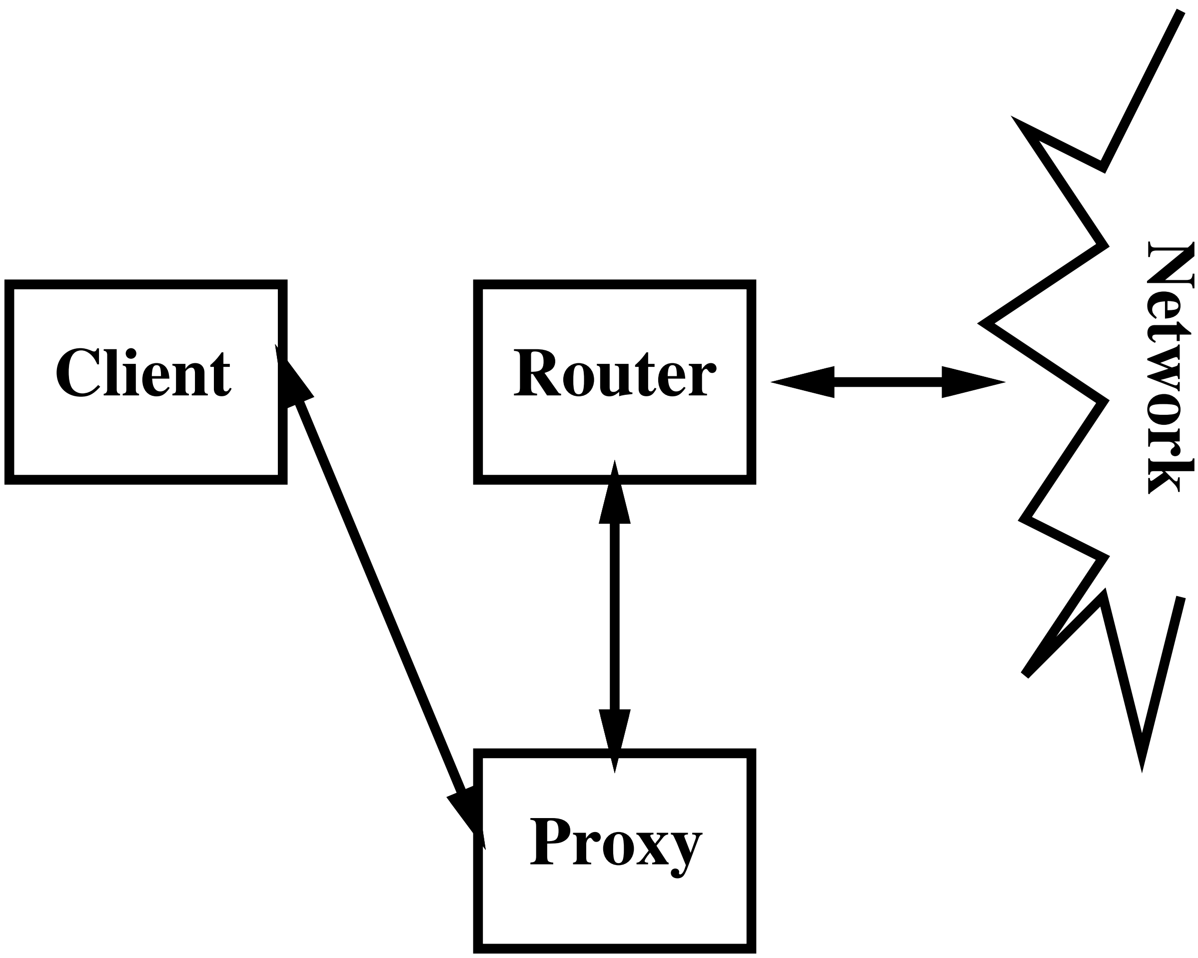
Capacity - ideal or effective?

Proxy Caches



Proxy Mechanisms

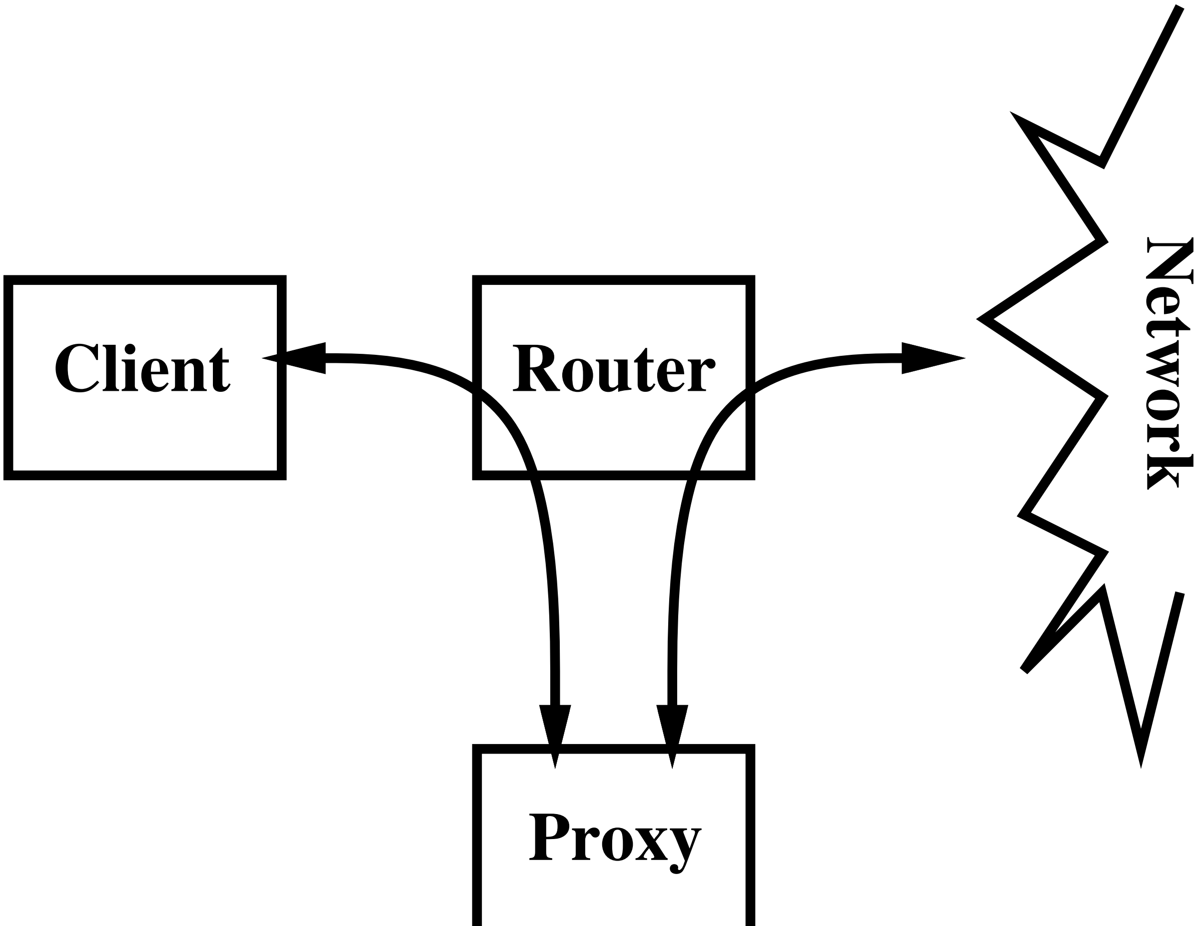
Forward mode



needs browser support

Proxy Mechanisms

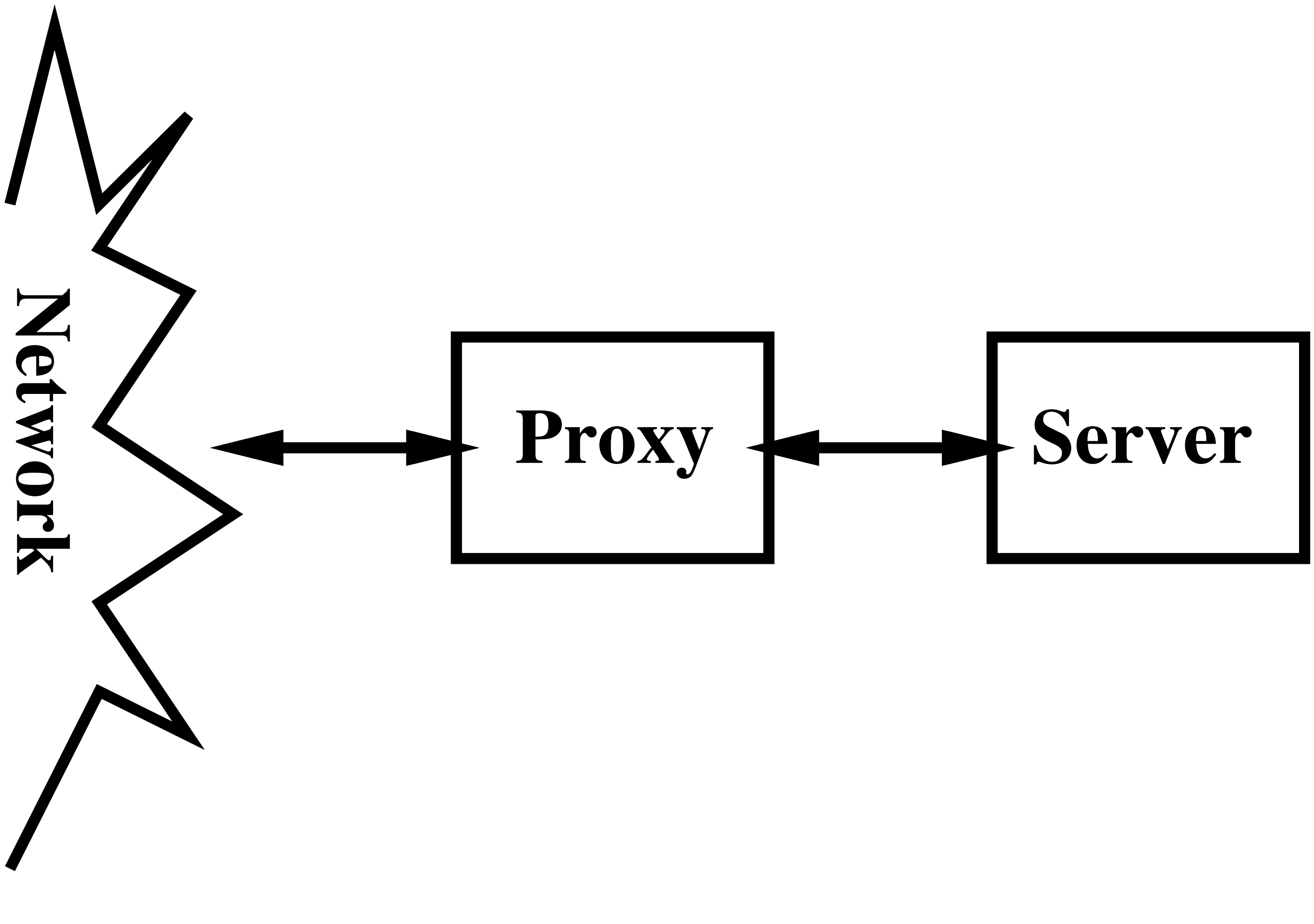
Transparent mode



needs router support

Proxy Mechanisms

Reverse mode



only server-side

Implications

Benefits

- fewer compulsory misses
- larger effective capacity

Drawbacks

- application-aware
- protocol-aware
- single point of failure
- possible overload

CERN Proxy

Fork per request

URL space = filename

Benefits:

- simple
- good "enough"

drawback:

- slow

Harvest Proxy

Single-process event-driven

URL space $\longrightarrow$ file number

Benefits:

- faster
- available
- available
- well-publicized

Later becomes Squid

Caching Issues

Correctness

- heuristics
- protocol changes

Performance

- different application behavior
- customized OS

Correctness

What can be cached

- static content
- public

How long can we cache

- specified expiration
- last modified time
- guess/default

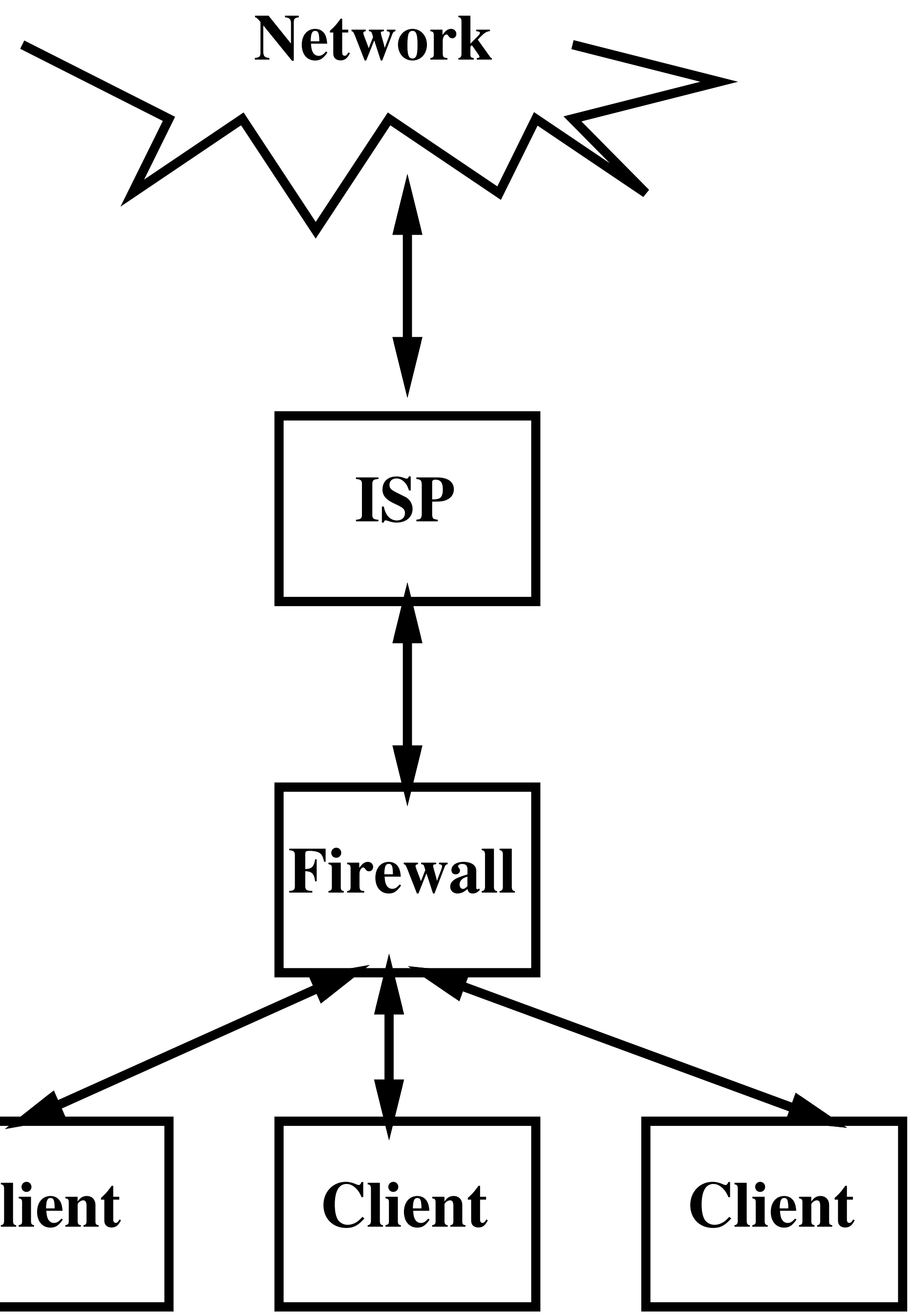
Performance

Where should we cache?

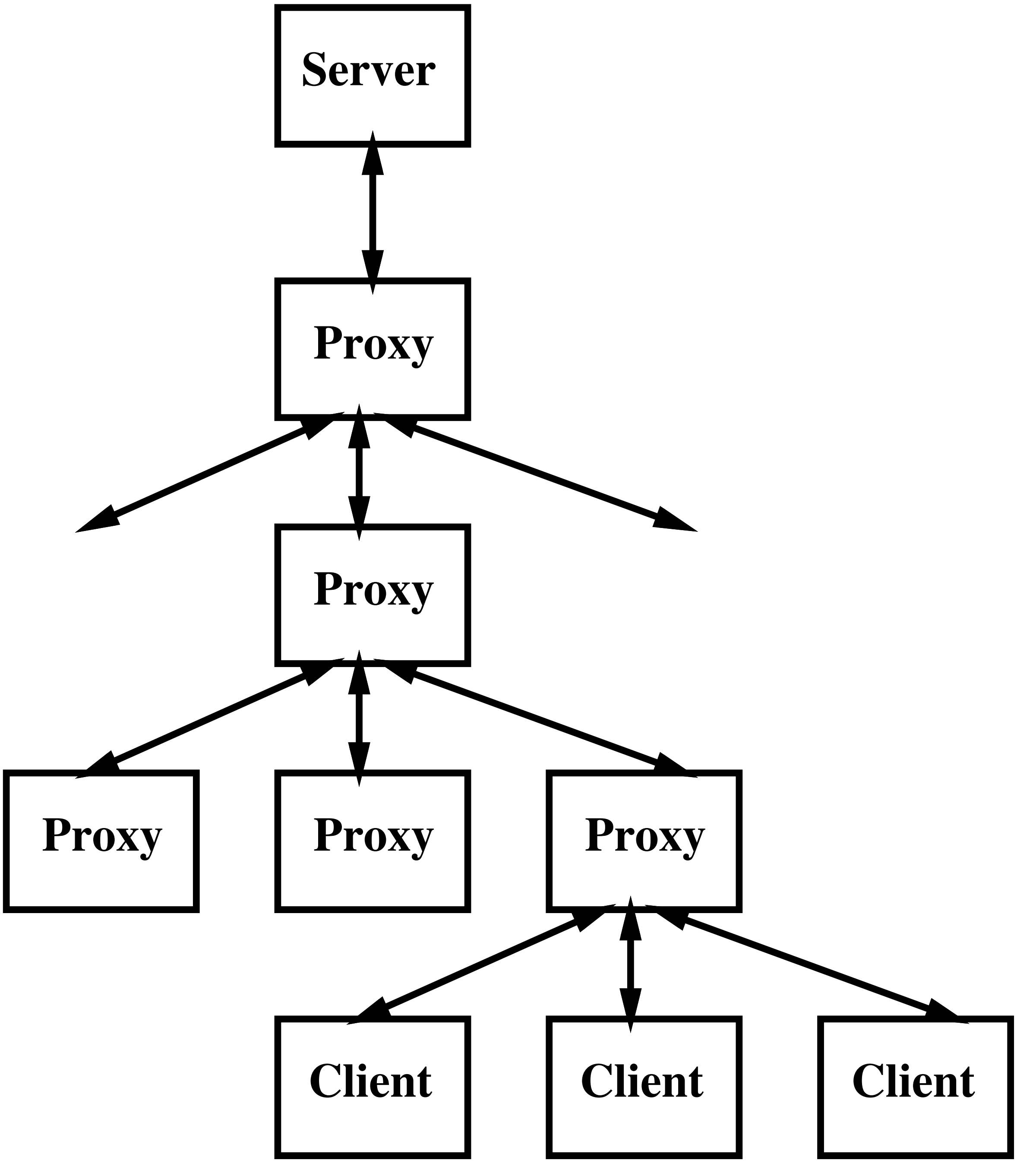
What should we cache?

How should we cache?

Where should we cache?



Cache Hierarchy



Cache Hierarchy

Lowest level supports clients

Upper levels support other proxies

Hits satisfied via neighbors

Misses go up hierarchy

Finding Files

Query neighbors

Exchange directories

Route deterministically

What Should Be Cached?

Files, not pages

Whole files only

Infinite document set

Cache Replacement

Files

- size
- frequency
- last use
- miss cost

Replacement Policies

Utility functions

- # of misses
- bytes transferred
- average latency
- perceived latency

How to Cache

Three main goals

- Reduce seeks
- Reduce seeks
- Reduce seeks

What Causes Seeks

Reading

Writing

Metadata

How To Reduce Seeks

Index metadata

Cache in main memory

Reorganize small accesses

Summary

Web has nonuniform access

Caching improves performance
- when locality exists
- when designed well

Cache organizations vary
- internal
- topology

Subject of ongoing research