



# **DHTTP: An Efficient and Cache-Friendly Transfer Protocol for the Web**

**IEEE/ACM Transactions on Networking  
December 2004**

# Introduction (1/3)

- The use of Web proxies
  - Advantages:
    - May saving file retrieval time
    - Relieving server load
  - Disadvantages:
    - Not really server-client end-to-end connection
    - Sometimes proxy may be a bottleneck
    - Cache maintenance

# Introduction (2/3)

- Two main features of current HTTP
  - Persistent connection
    - Amortizing the TCP setup overhead
  - Pipelining transmission
    - A client may send multiple requests over the same connection without waiting for responses
    - The server will send a stream back

# Introduction (3/3)

- These two features introduce new performance penalties
  - Persistent connection
    - Throughput degradation significant while the number of open connections increasing
  - Pipelining transmission
    - Server must send responses in their entirety and in the same order as the order of the requests in the pipeline
    - This constraint causes head-of-line delays when a slow response holds up all other responses in the pipeline

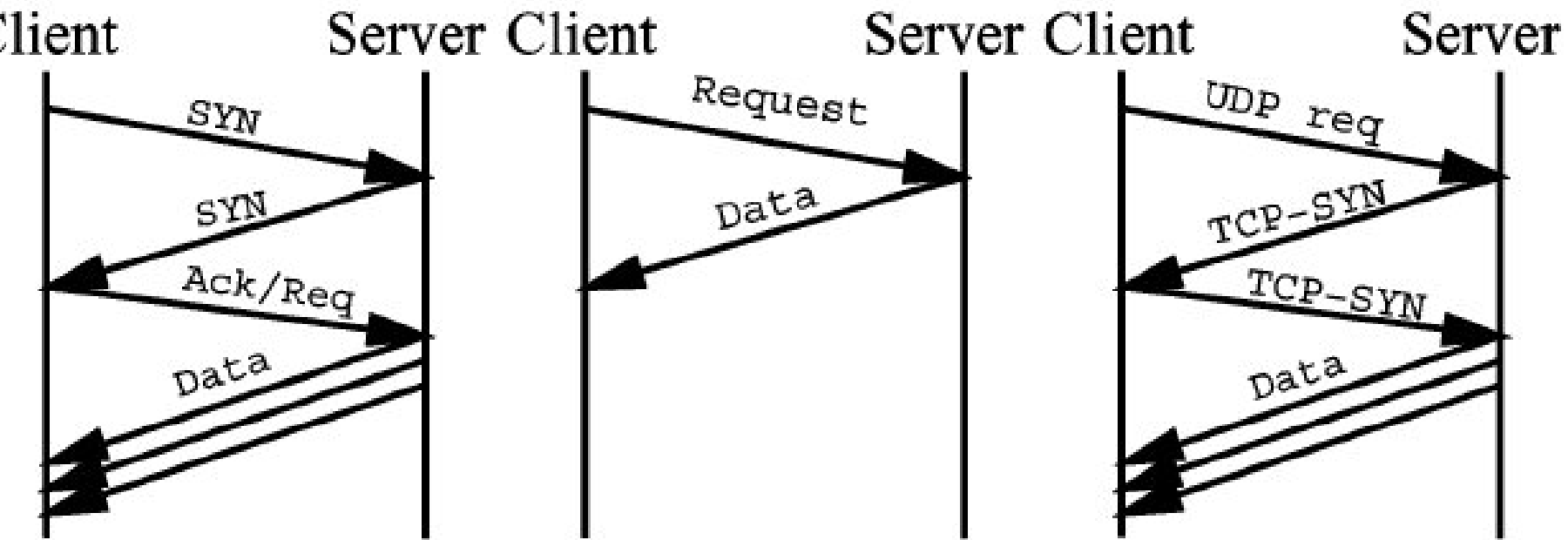
# Main Ideas

- Splitting Web traffic between UDP and TCP
  - The client sends requests by UDP
  - The server sends its response over UDP or TCP
    - Size of response
    - Network condition
- Server establishes the connection back to the client when choosing TCP

# Dual-Transport HTTP Protocol (DHTTP)

- Both Web clients and servers listen on two ports
  - One for UDP and the other for TCP
    - Like UDP channel and TCP channel
    - The client sends requests through UDP channel
    - The server sends its response over UDP channel or TCP channel

# Message Exchange

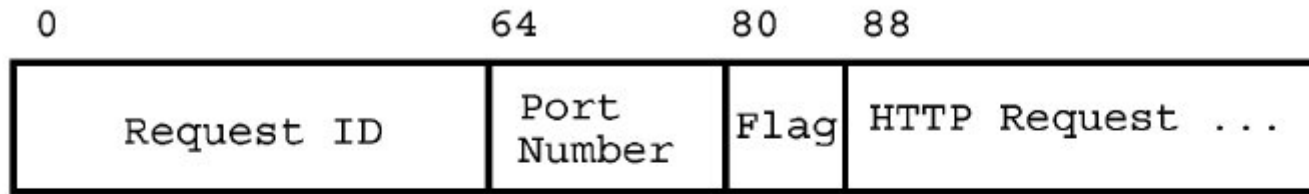


Current HTTP

DHTTP over UDP

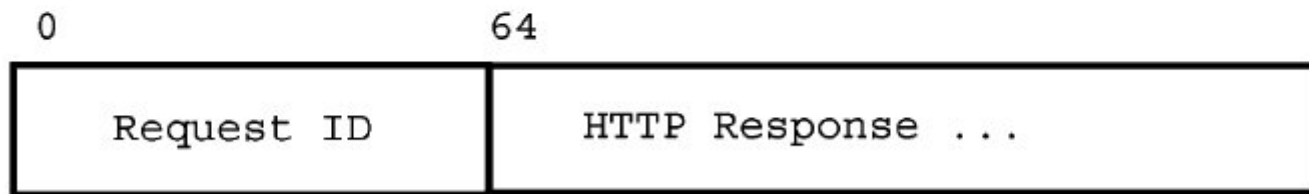
DHTTP over TCP

# DHTTP message formats



Request

(a)



Response

(b)



# DHTTP

- Reliability and non-idempotent requests
  - DHTTP stipulates that a client may resend a UDP request
  - DHTTP delegates the non-idempotent requests to TCP channel
- Congestion control
  - DHTTP leverages TCP by requiring that responses to any resent requests be sent over TCP

# DHTTP

- Choosing a channel

- Based on

- The response size
- Network condition

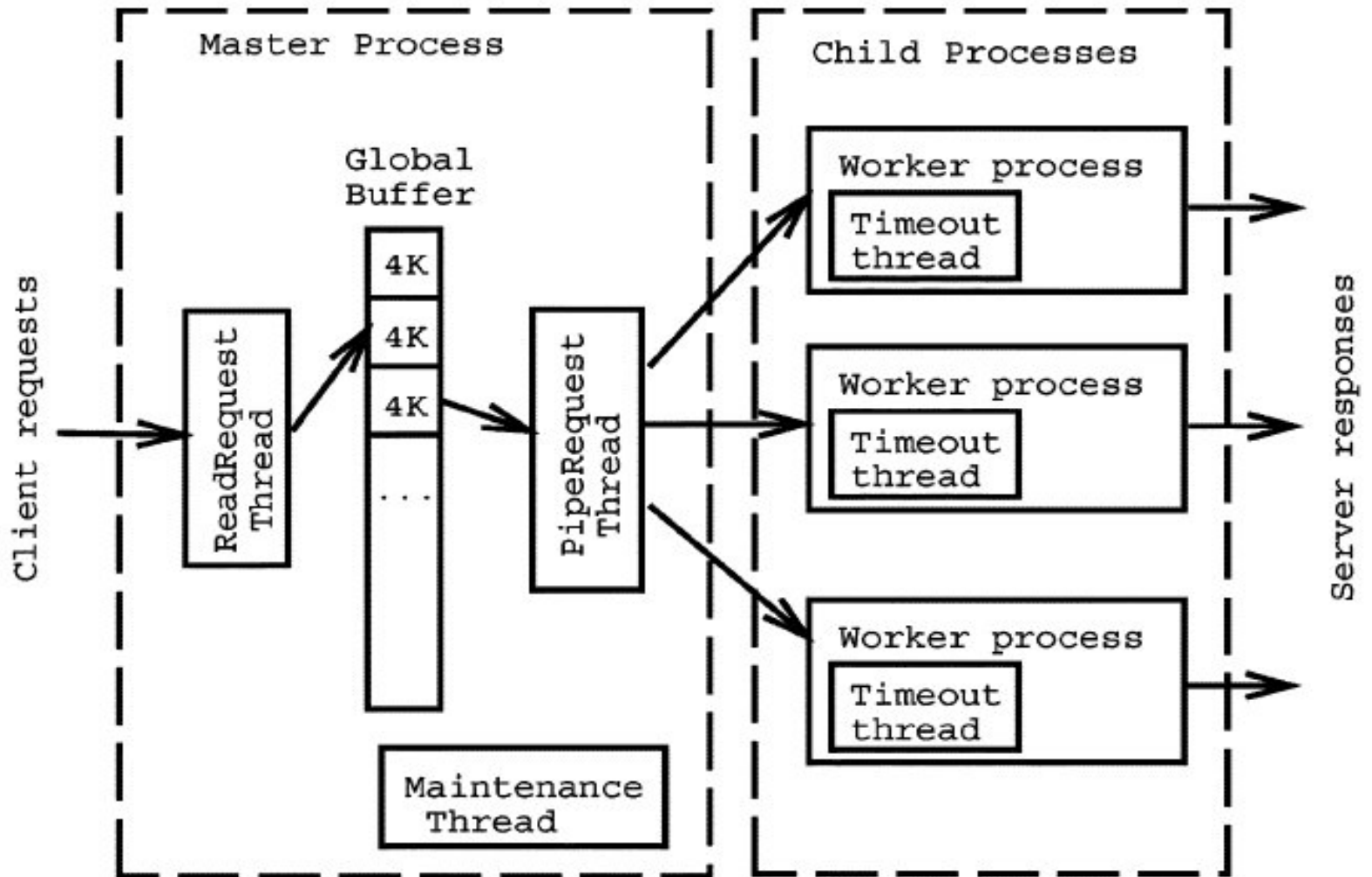
- Maintaining a “fresh requests counter” and a “resent requests counter”
- Loss threshold “ $L$ ”
- Size threshold “ $S$ ”

- 1) Choose TCP for all large responses, i.e., whose size exceeds  $S$ , as well as for all resent requests.
- 2) If the ratio of resent request counter to fresh request counter exceeds  $L$ , enter a “high-loss” mode, else enter a “low-loss” mode.
- 3) In the low-loss mode, choose UDP for all small responses, i.e., those below the size threshold  $S$ .
- 4) In the high-loss mode, choose TCP for the  $1 - L$  fraction of small responses and UDP for the remaining  $L$  small responses.

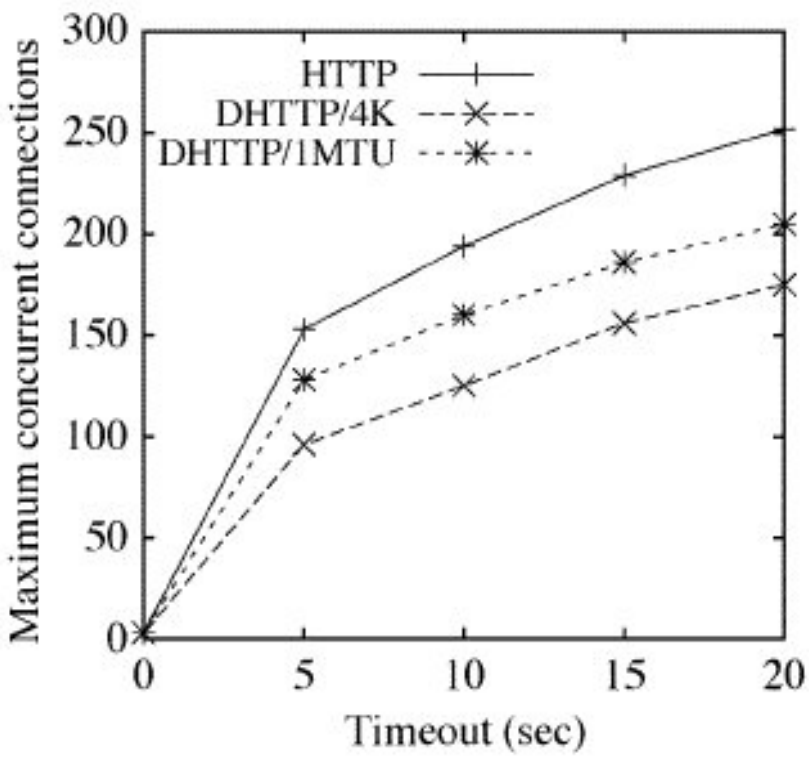
# DHTTP

- DHTTP and interception caches
  - A DHTTP interception cache will intercept only requests sent over UDP and pass through any request using TCP
  - The client is aware it speaks with the cache
- DHTTP using UDP channel to transmit short data
  - Relieving the overhead of TCP channel
  - Shortening the response time

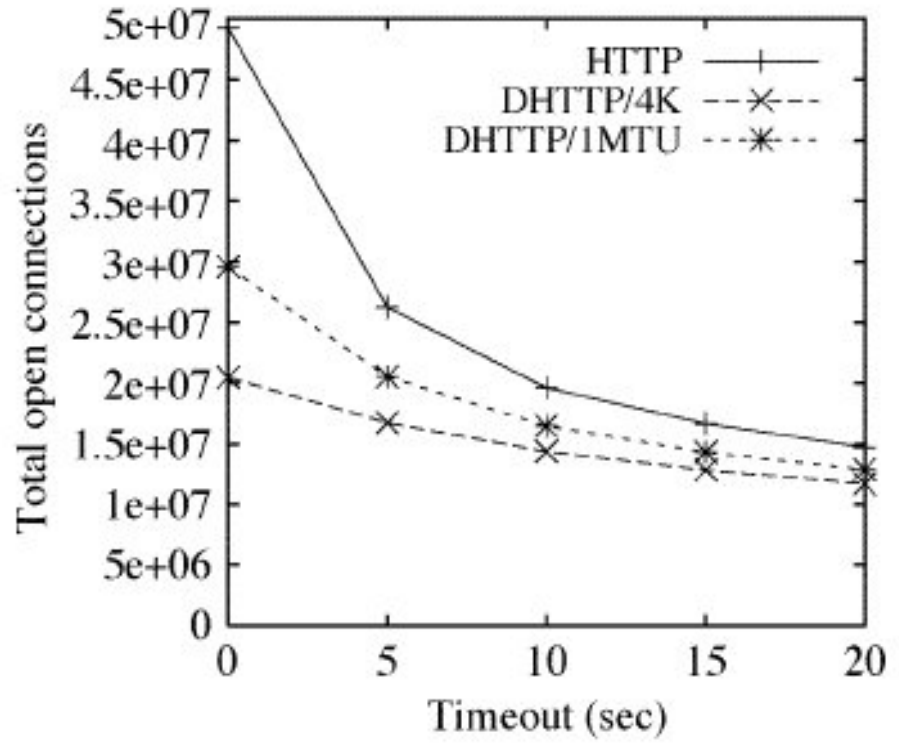
# DHTTP Server Architecture



# Number of TCP Connections at a Server with Three Connections Per Client

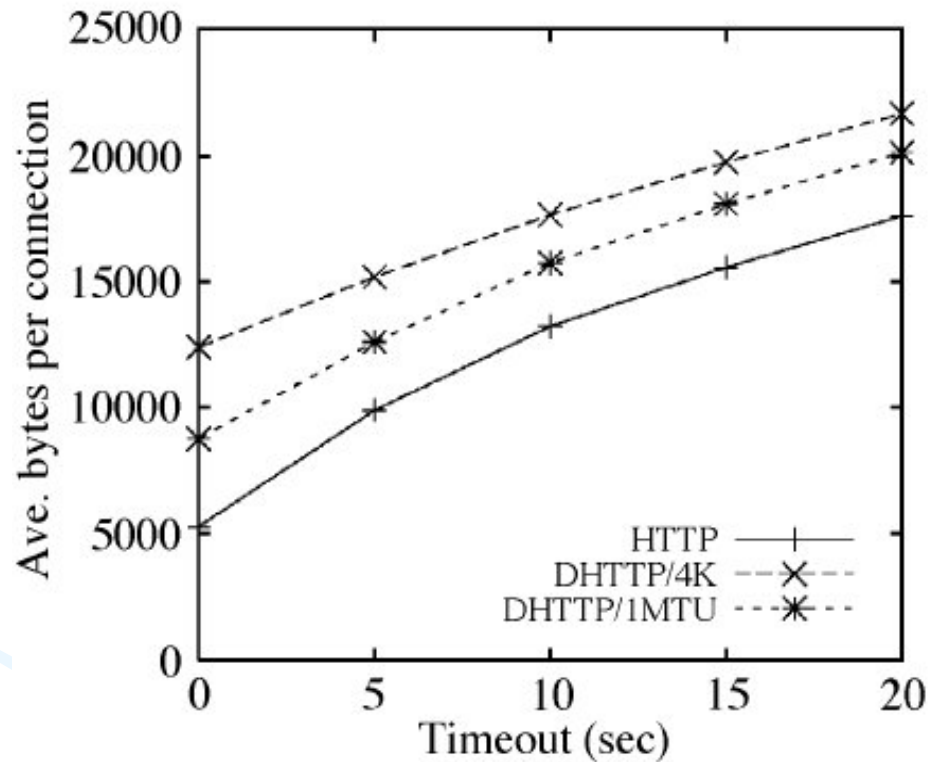


(a)

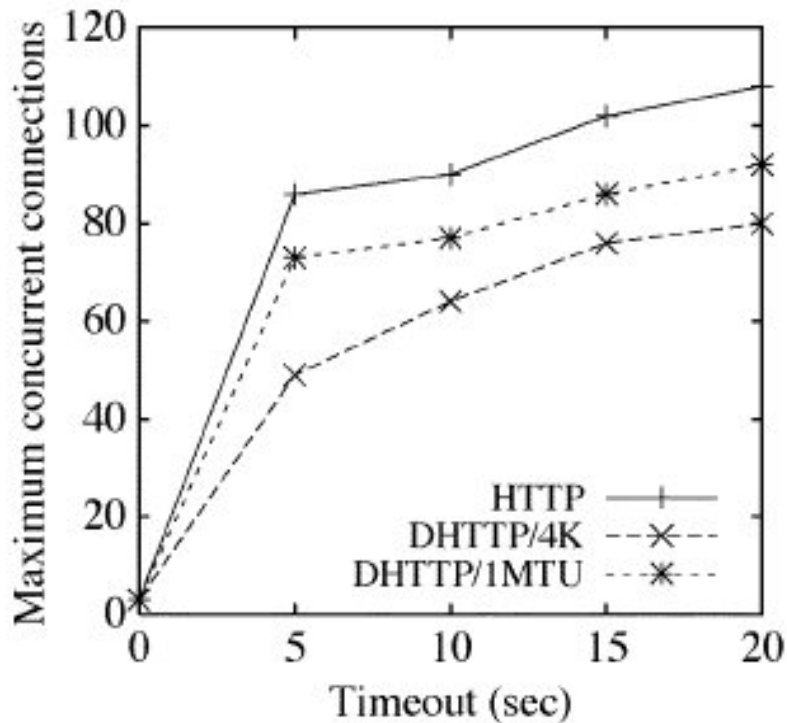


(b)

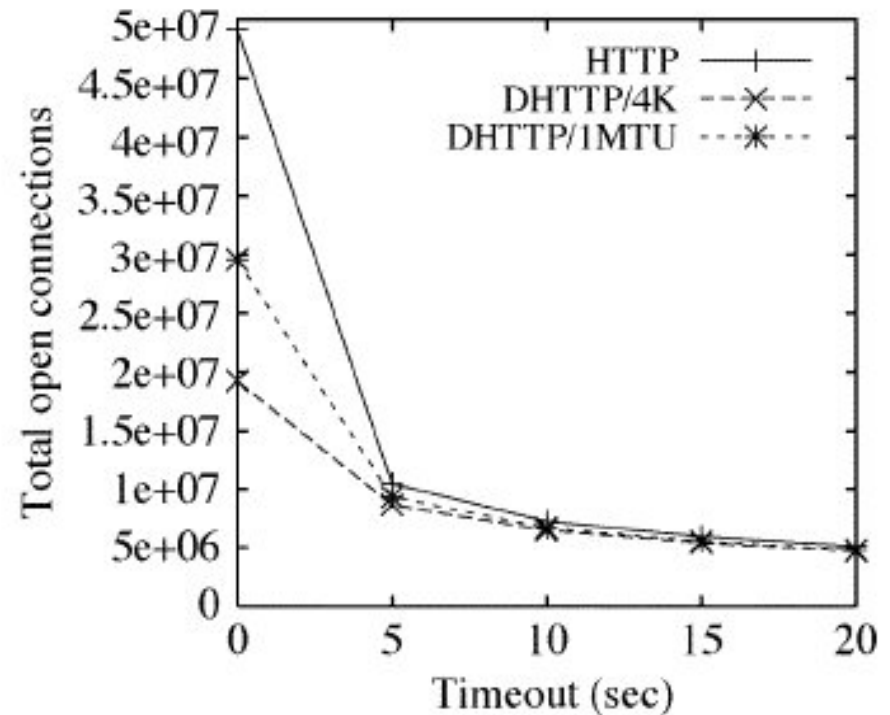
# Connection Utilization with Three Connections Per Client



# Number of TCP Connections at a Server with One Connections Per Client

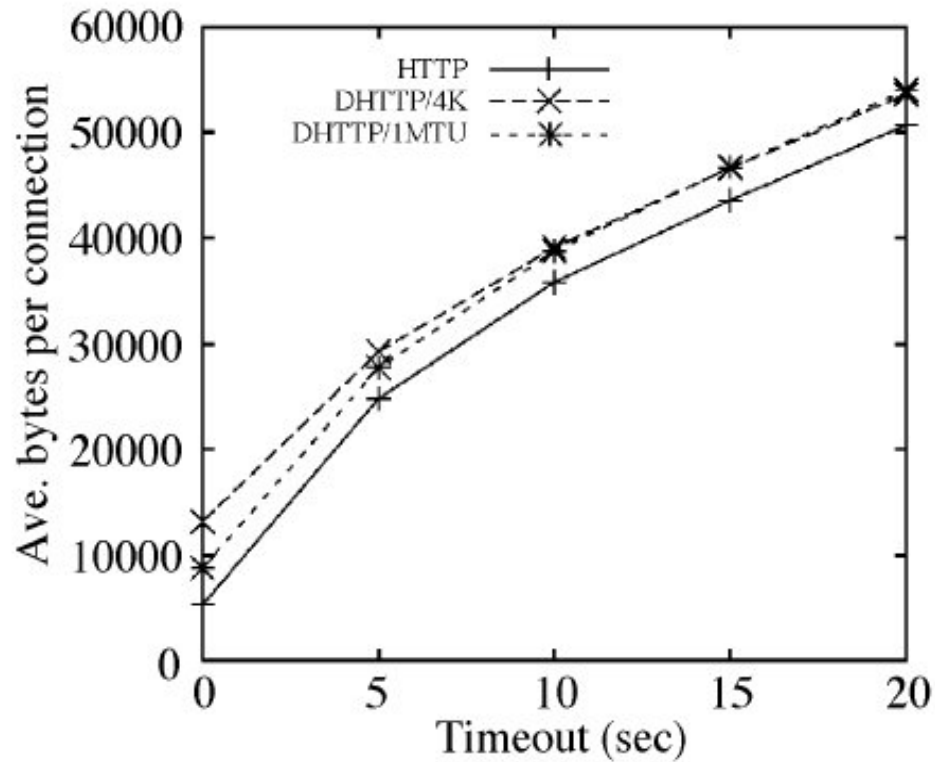


(a)



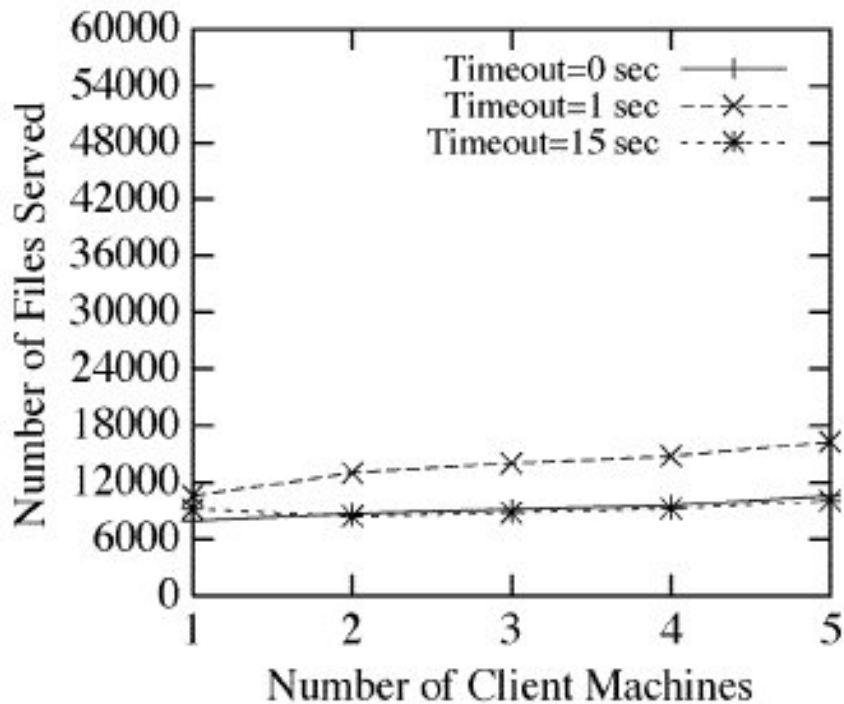
(b)

# Connection Utilization with One Connections Per Client

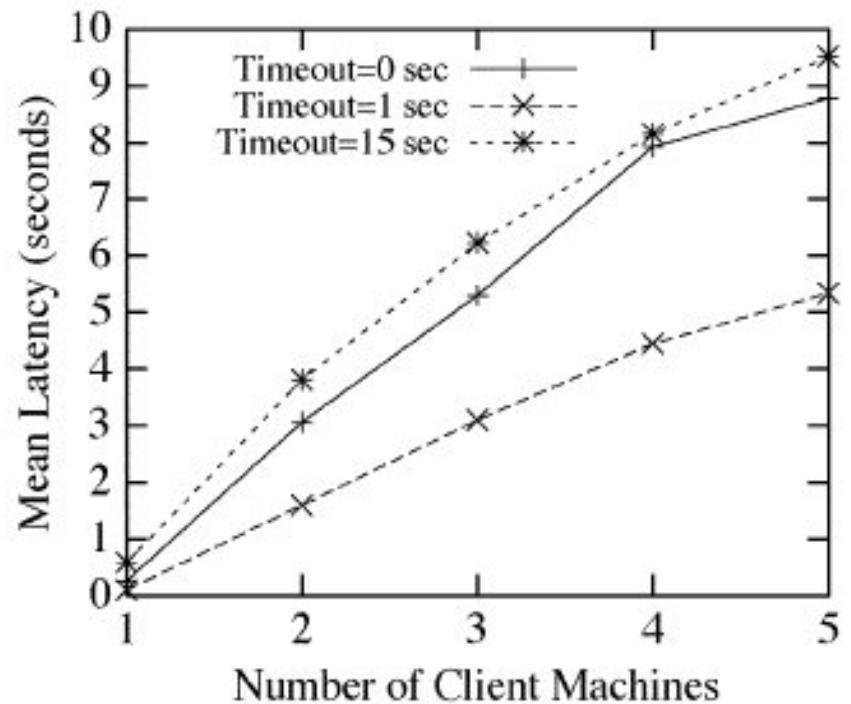




# Testing Results

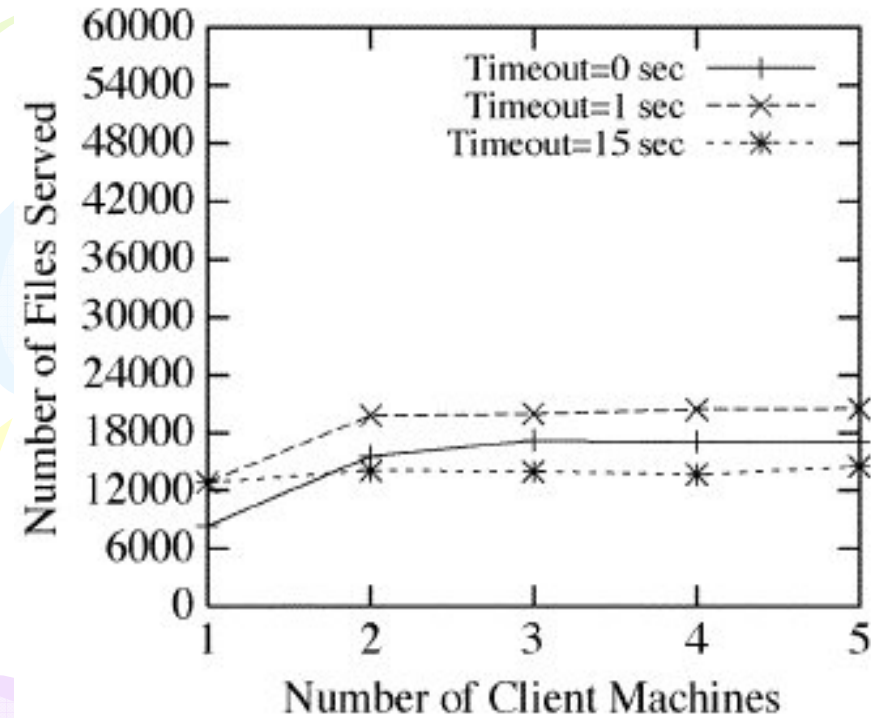


(a)

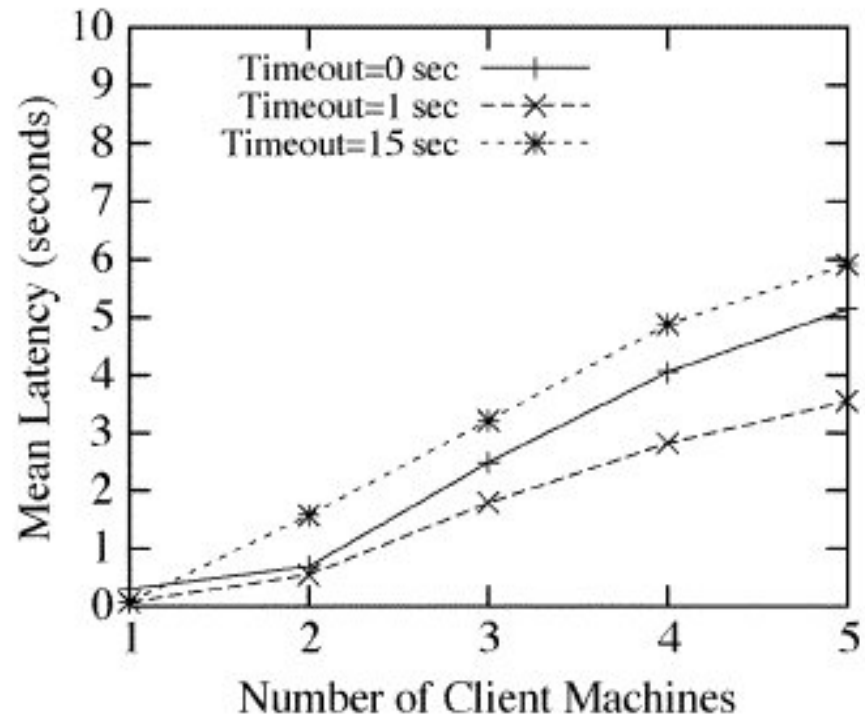


(b)

# Testing Results

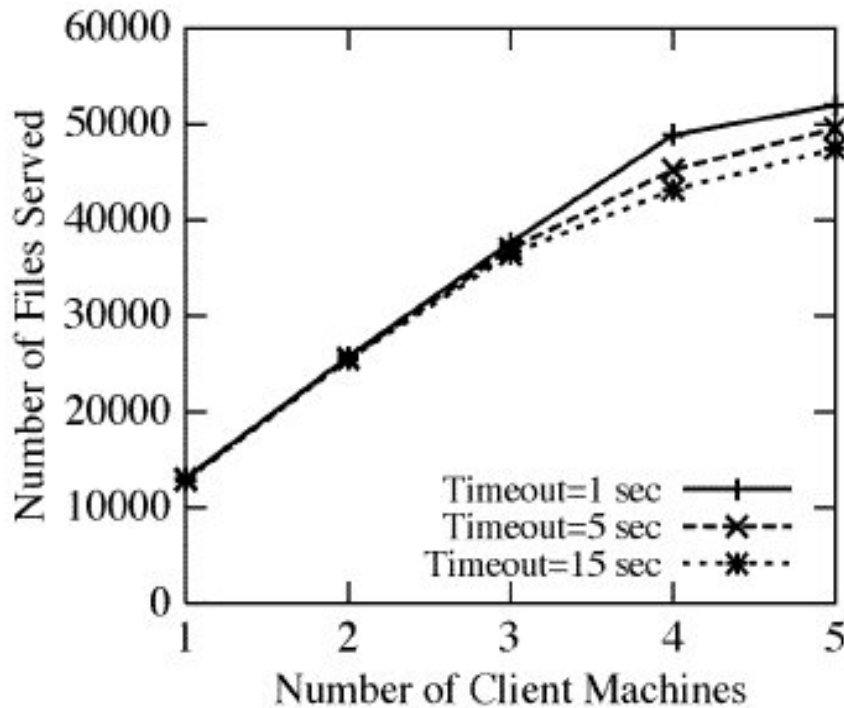


(c)

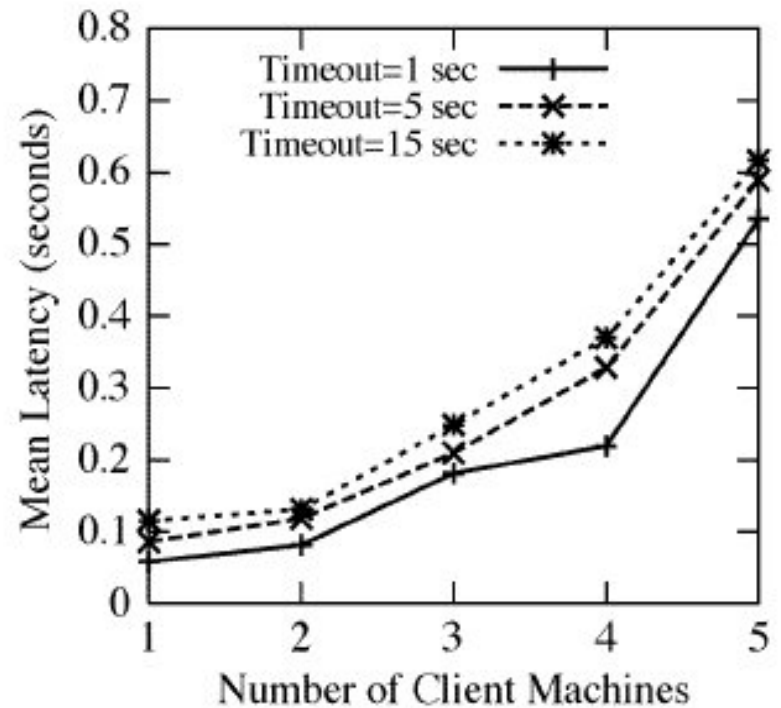


(d)

# Server Performance

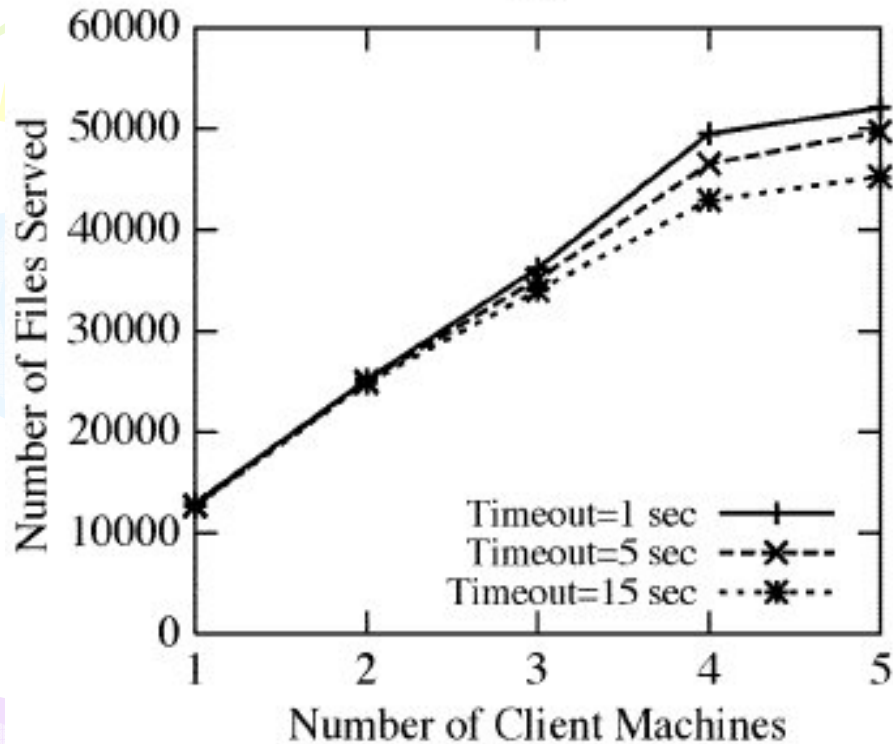


(a)

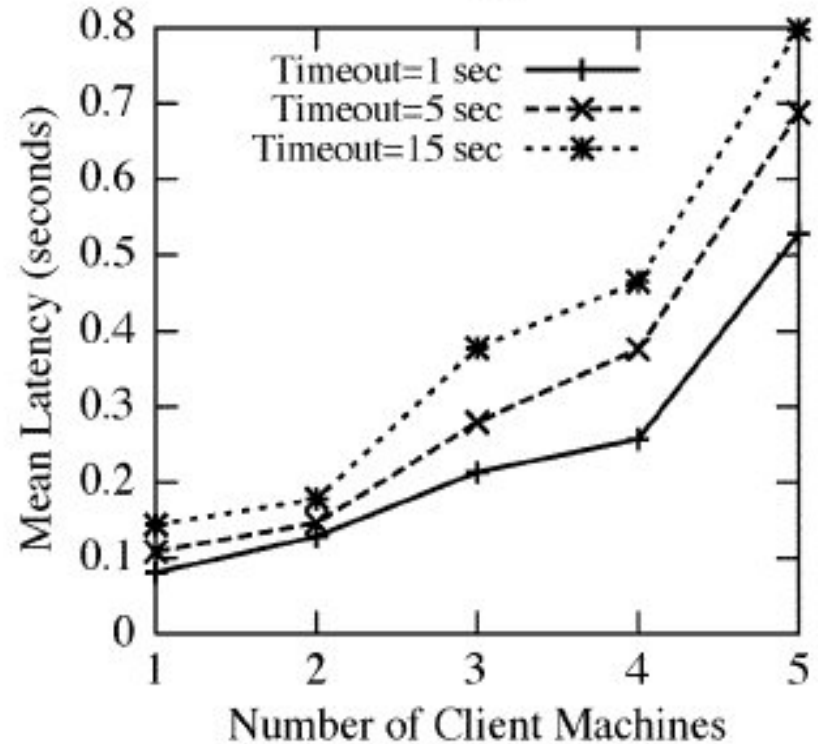


(b)

# Server Performance

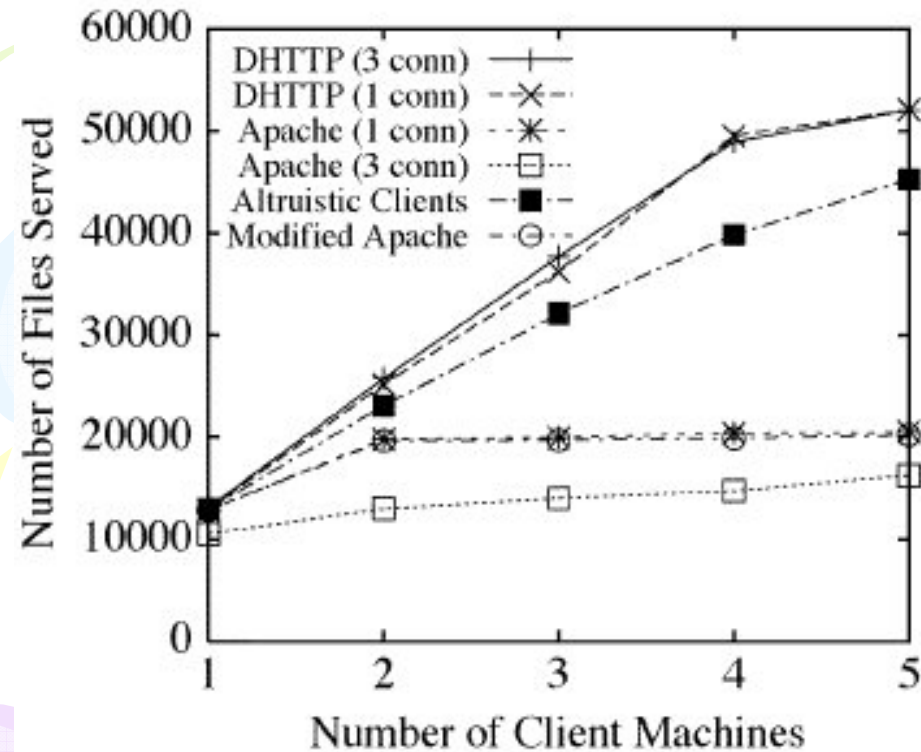


(c)

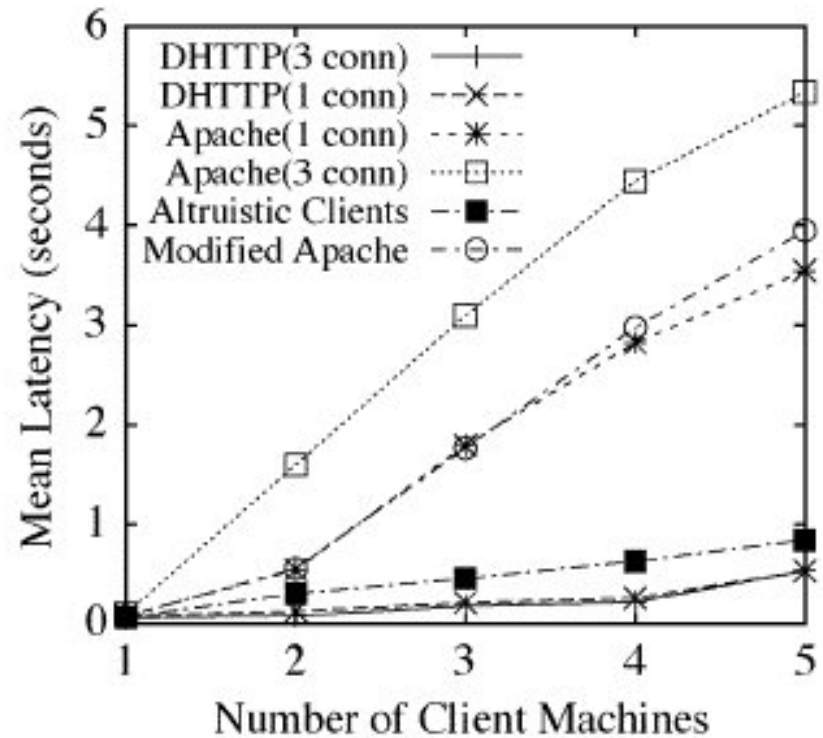


(d)

# Performance Comparison

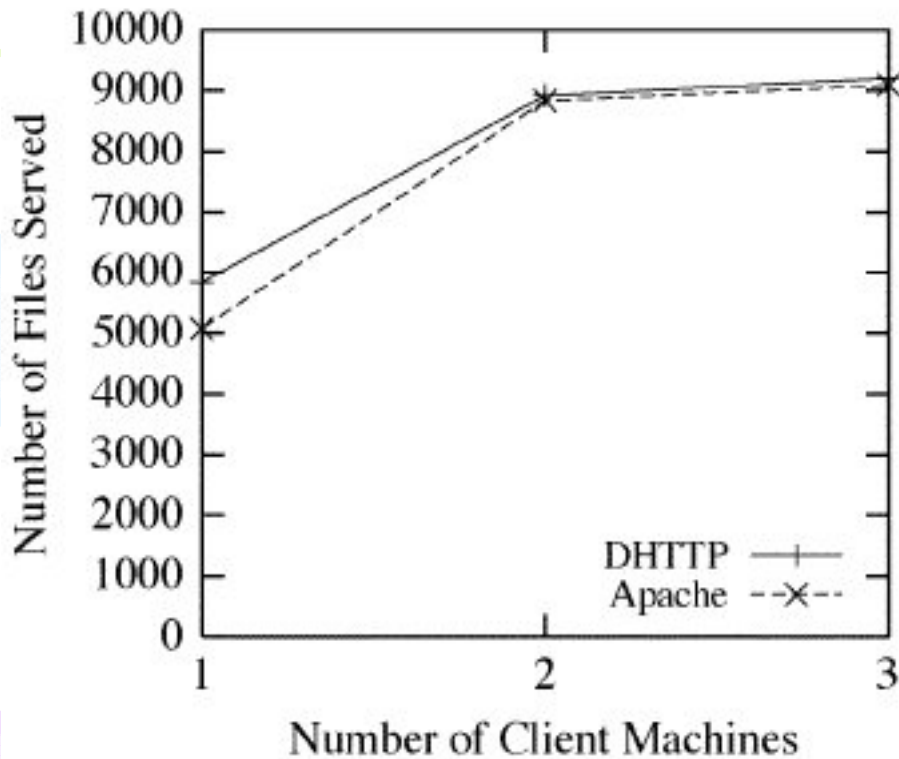


(a)

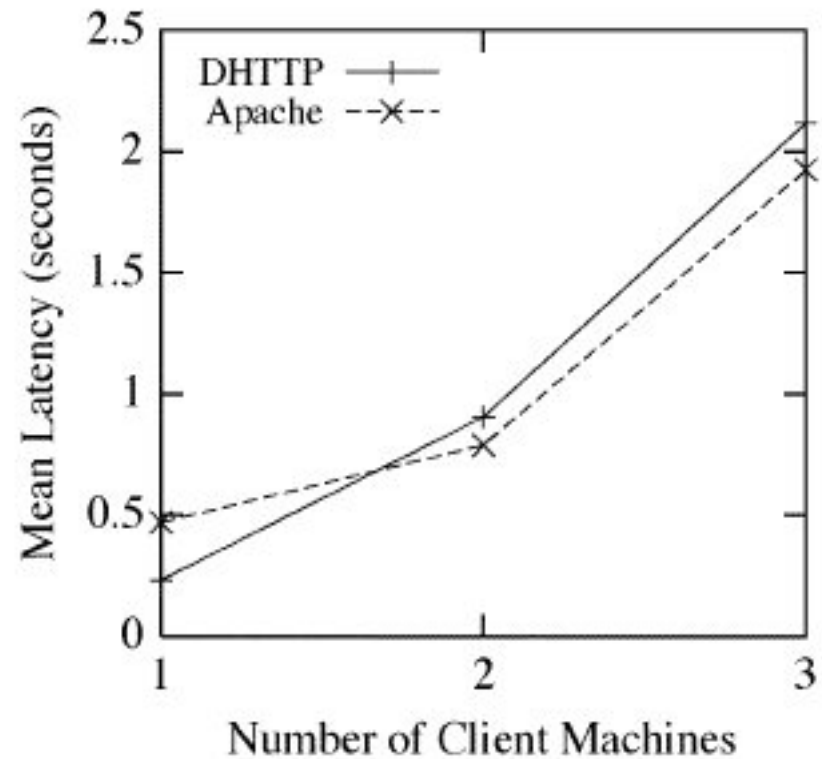


(b)

# Performance Comparison

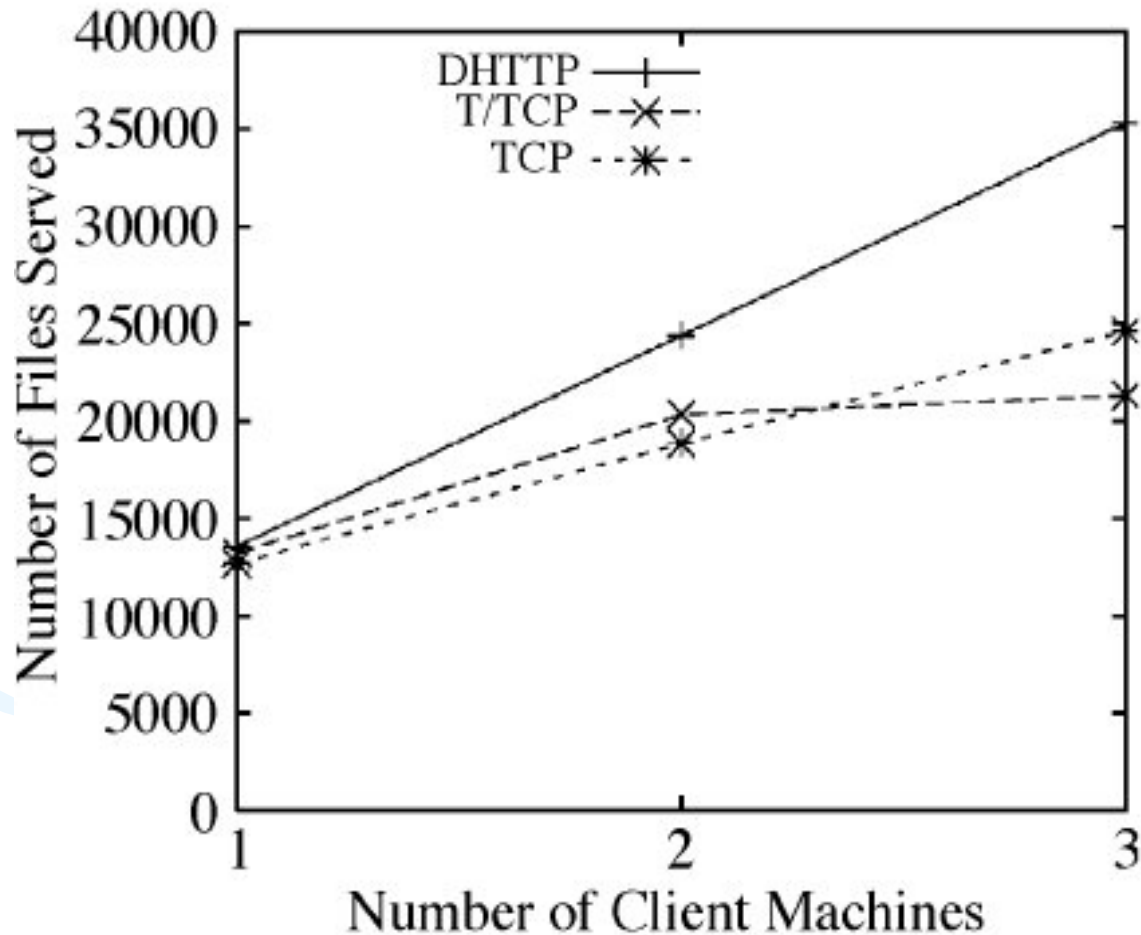


(a)



(b)

# Performance Comparison



# Conclusion

- DHTTP splits Web traffic between UDP and TCP
  - Clients send requests by UDP
  - Server sends its response over UDP or TCP
- A DHTTP interception cache will intercept only requests sent over UDP and pass through any request using TCP
- Server establishes the connection back to the client,