Reducing Web Latency Using Reference Point Caching

Authors: G. P. Chandranmenon and G. Varghese

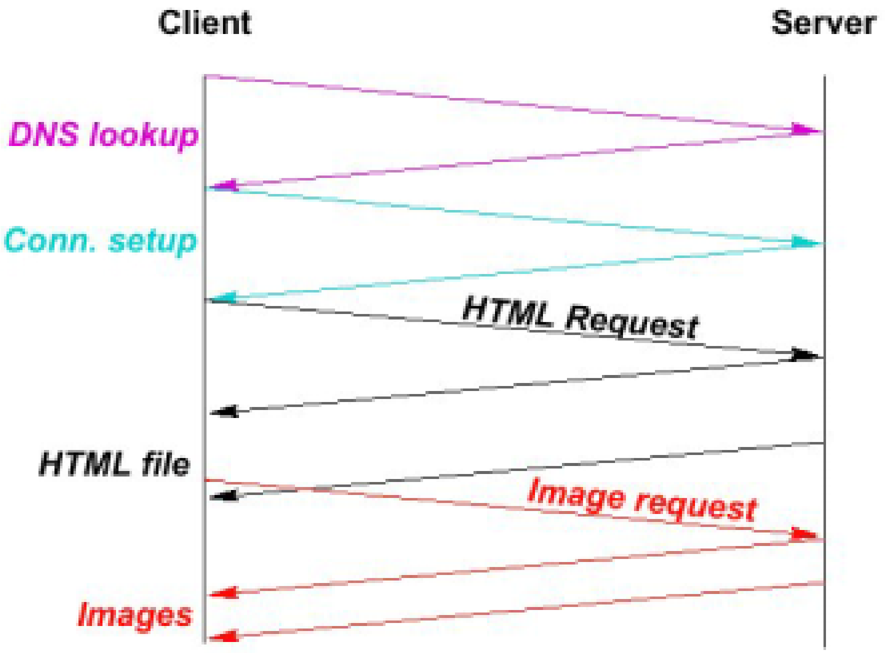
INFOCOM 2001

Outline

- Problem Statement
- The Proposed Scheme
  - Reference Point Caching
    - Caching IP Addresses
    - Caching Documents
- Conclusion
- Discussion

Problem Statement

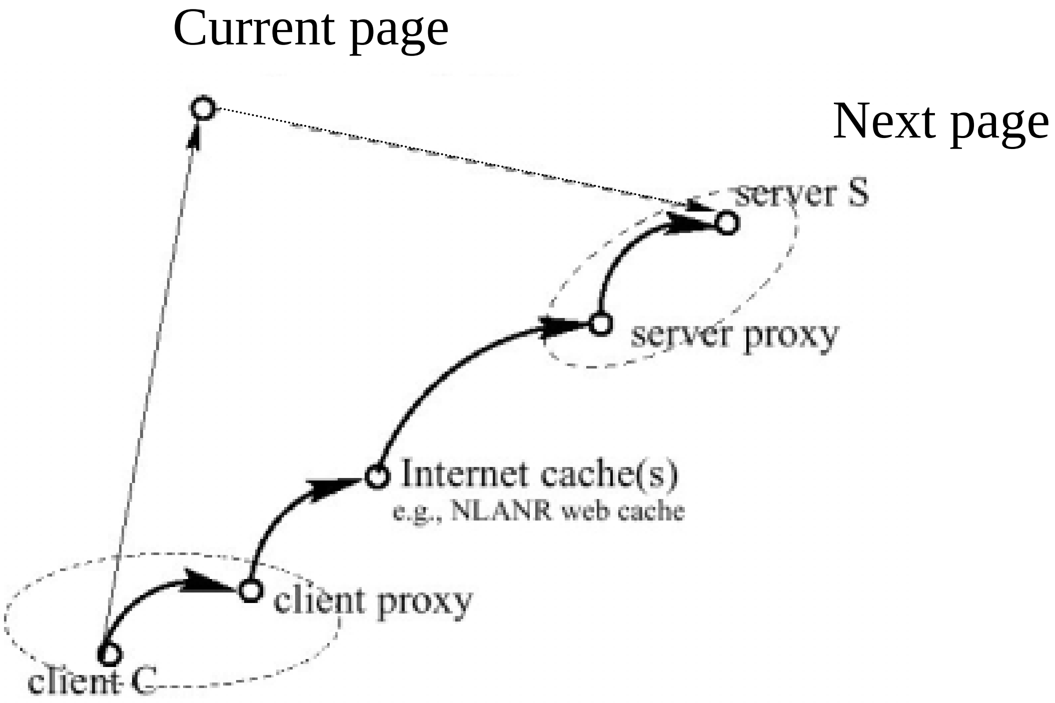
- How to reduce web access latencies?



•Steps in a typical web access.

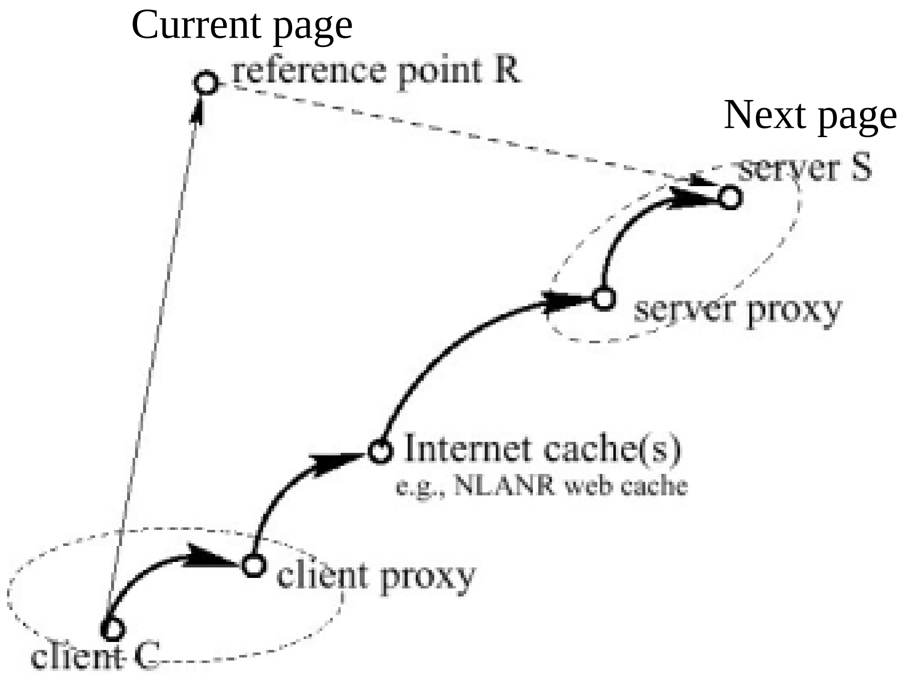
Problem Statement (cont.)

- Waste much time on DNS-lookup and connection setup.

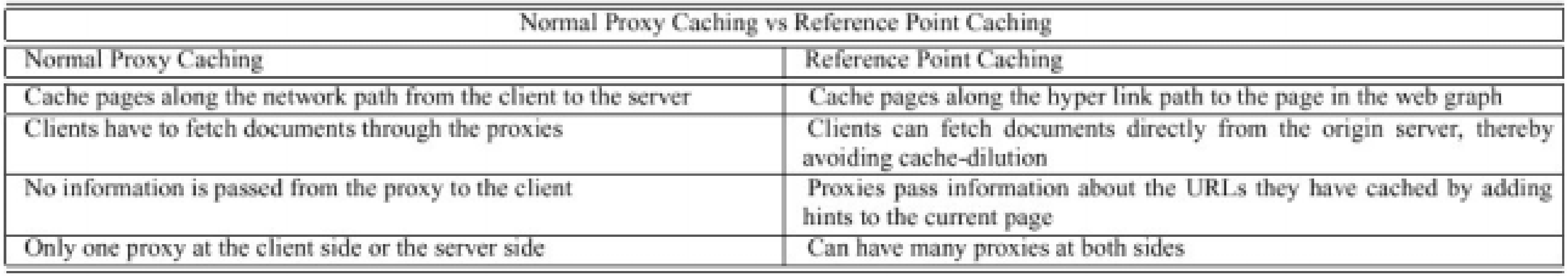


Main Ideas

- R(called the reference point) has a link refer to a page on S.
- The reference point R is allowed to have a cached copy of the page at S.
- Or R can return the IP address of S to Client C.



Comparison



Normal Proxy Caching vs Reference Point Caching

Normal Proxy Caching	Reference Point Caching
Cache pages along the network path from the client to the server	Cache pages along the hyper link path to the page in the web graph
Clients have to fetch documents through the proxies	Clients can fetch documents directly from the origin server, thereby avoiding cache-dilution
No information is passed from the proxy to the client	Proxies pass information about the URLs they have cached by adding hints to the current page
Only one proxy at the client side or the server side	Can have many proxies at both sides

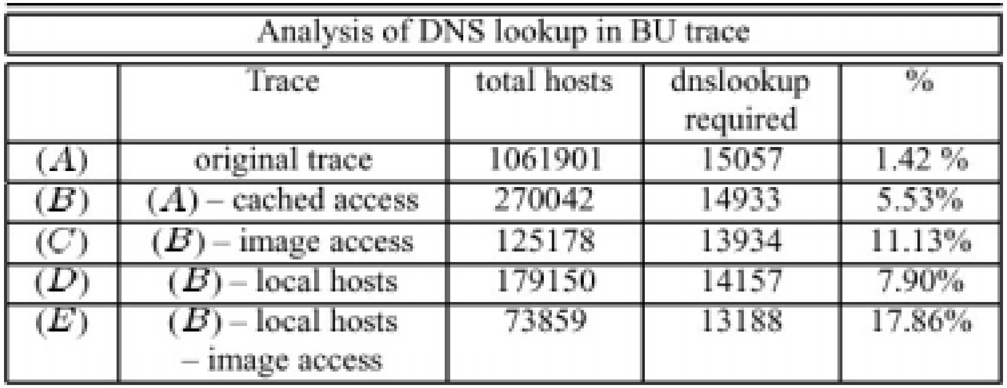
Design Issues

- The web page must include extra information
  - The cache information about the link page.
  - The link’s IP address and its valid time.
- Using MIME headers and HTML tags
  - The servers, proxies and clients that don’t understand the modifications won’t be affected.
- The browser also have the ability to deal with the extra information.

Caching IP Address

- We should find out:
  - The average DNS-lookup time
  - The typical life time of DNS cache entries
- File overheads
  - Increase the file size by 24 bytes for every unique host name in a page.
- The idea can be applied to other applications that do DNS lookups.

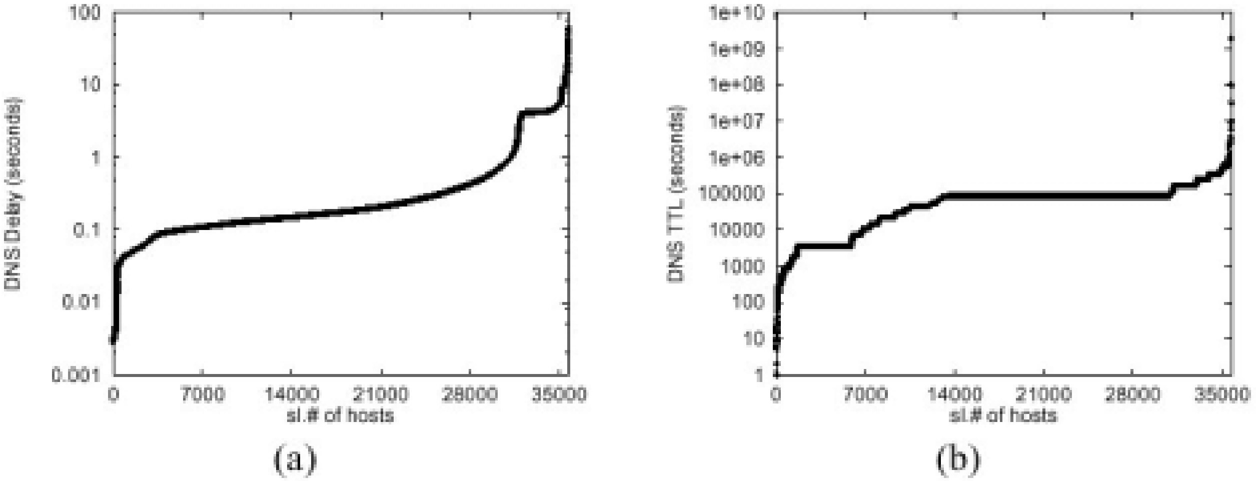
Trace Result of DNS-Lookup(1/2)



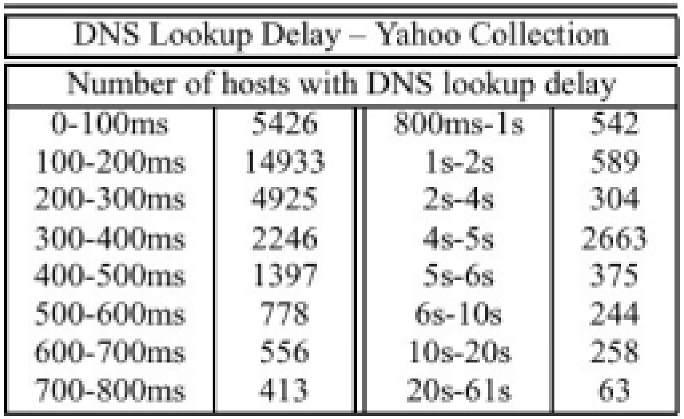
Analysis of DNS lookup in BU trace

	Trace	total hosts	dnslookup required	%
(A)	original trace	1061901	15057	1.42 %
(B)	(A) – cached access	270042	14933	5.53%
(C)	(B) – image access	125178	13934	11.13%
(D)	(B) – local hosts	179150	14157	7.90%
(E)	(B) – local hosts – image access	73859	13188	17.86%

Trace Result of DNS-Lookup(2/2)



(a) (b)

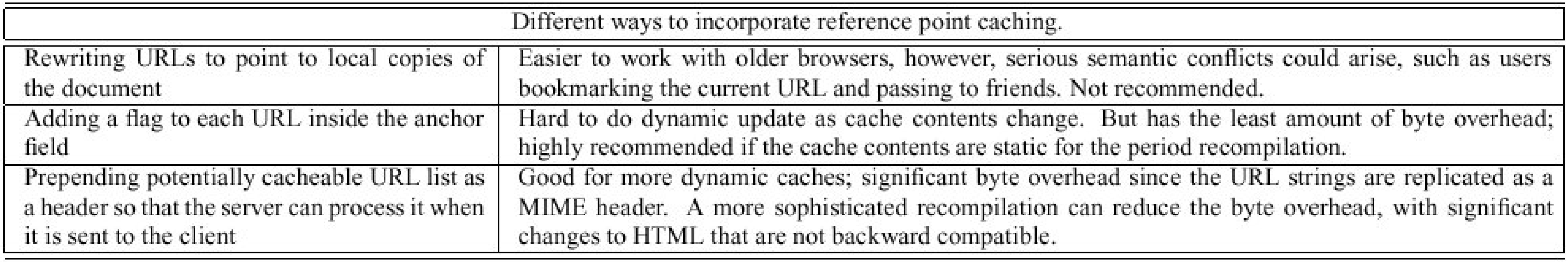


DNS Lookup Delay – Yahoo Collection			
Number of hosts with DNS lookup delay			
0-100ms	5426	800ms-1s	542
100-200ms	14933	1s-2s	589
200-300ms	4925	2s-4s	304
300-400ms	2246	4s-5s	2663
400-500ms	1397	5s-6s	375
500-600ms	778	6s-10s	244
600-700ms	556	10s-20s	258
700-800ms	413	20s-61s	63

Caching Documents

- With this mechanism, reference point caching enables every web server to be a potential proxy cache.
  - It avoids single point bottlenecks at a proxy.
  - Each web server can choose to cache and serve only the documents it wants.

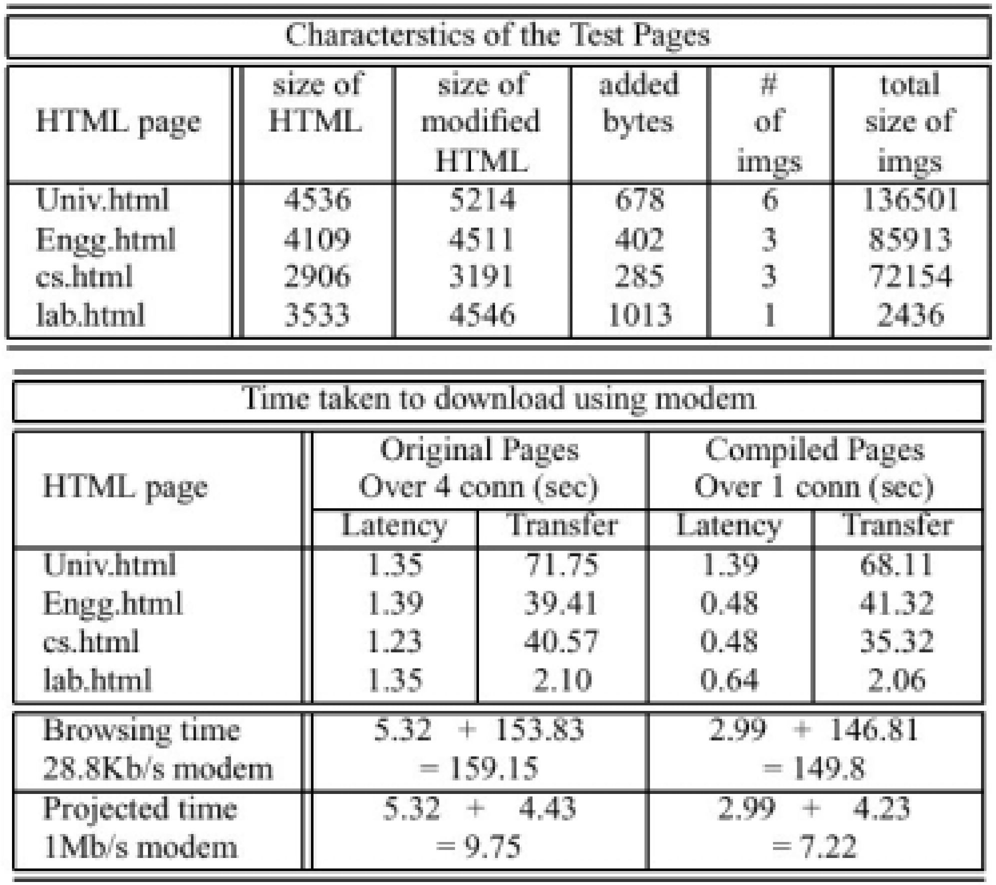
Incorporating into HTTP



Different ways to incorporate reference point caching.

Rewriting URLs to point to local copies of the document	Easier to work with older browsers, however, serious semantic conflicts could arise, such as users bookmarking the current URL and passing to friends. Not recommended.
Adding a flag to each URL inside the anchor field	Hard to do dynamic update as cache contents change. But has the least amount of byte overhead; highly recommended if the cache contents are static for the period recompilation.
Prepending potentially cacheable URL list as a header so that the server can process it when it is sent to the client	Good for more dynamic caches; significant byte overhead since the URL strings are replicated as a MIME header. A more sophisticated recompilation can reduce the byte overhead, with significant changes to HTML that are not backward compatible.

Experiment Results

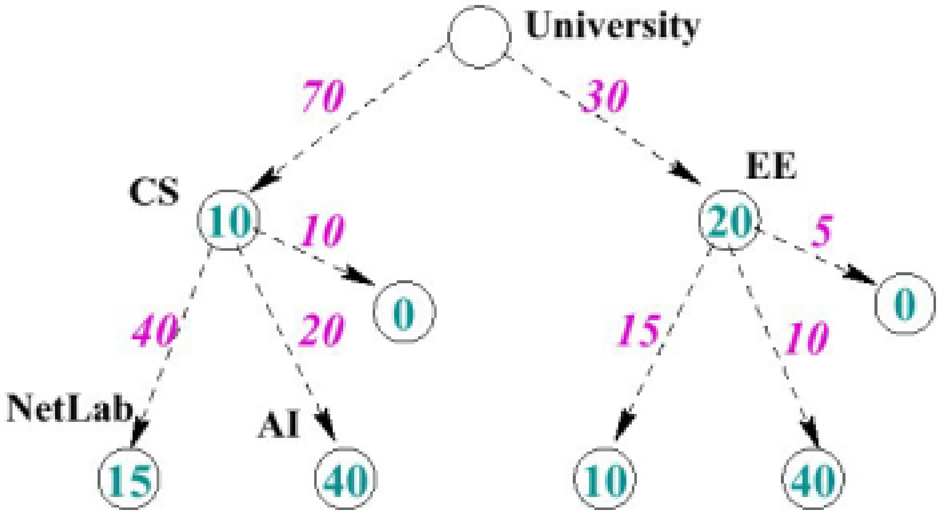


Characterstics of the Test Pages					
HTML page	size of HTML	size of modified HTML	added bytes	# of imgs	total size of imgs
Univ.html	4536	5214	678	6	136501
Engg.html	4109	4511	402	3	85913
cs.html	2906	3191	285	3	72154
lab.html	3533	4546	1013	1	2436

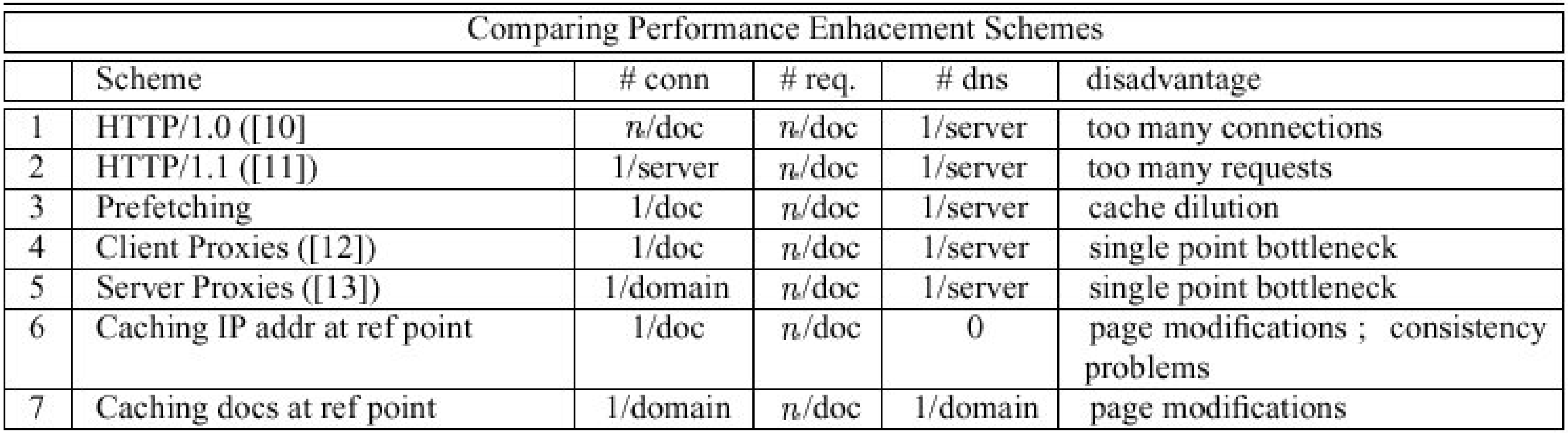
Time taken to download using modem				
HTML page	Original Pages Over 4 conn (sec)		Compiled Pages Over 1 conn (sec)	
	Latency	Transfer	Latency	Transfer
Univ.html	1.35	71.75	1.39	68.11
Engg.html	1.39	39.41	0.48	41.32
cs.html	1.23	40.57	0.48	35.32
lab.html	1.35	2.10	0.64	2.06
Browsing time 28.8Kb/s modem	5.32 + 153.83 = 159.15		2.99 + 146.81 = 149.8	
Projected time 1Mb/s modem	5.32 + 4.43 = 9.75		2.99 + 4.23 = 7.22	

Document Caching Policies

- Server side policies:
  - Document Size
  - Document Access Frequency
  - Available Memory and Disk Space
  - …



Summary(1/2)



Comparing Performance Enhacement Schemes					
	Scheme	# conn	# req.	# dns	disadvantage
1	HTTP/1.0 ([10]	n/doc	n/doc	1/server	too many connections
2	HTTP/1.1 ([11])	1/server	n/doc	1/server	too many requests
3	Prefetching	1/doc	n/doc	1/server	cache dilution
4	Client Proxies ([12])	1/doc	n/doc	1/server	single point bottleneck
5	Server Proxies ([13])	1/domain	n/doc	1/server	single point bottleneck
6	Caching IP addr at ref point	1/doc	n/doc	0	page modifications ; consistency problems
7	Caching docs at ref point	1/domain	n/doc	1/domain	page modifications

Summary(2/2)

- Avoiding DNS-lookup saves 100~300ms on the average, and sometimes on the order of seconds.
- Avoiding connection setup can save 240ms on the average.

Discussion

- The interaction between reference point caching and DNS-based load balancing scheme.
- Normal Proxy caches are still necessary.
- How about wireless environments?