University of North Carolina at Chapel Hill

What TCP/IP Protocol Headers Can Tell Us About the Web

Félix Hernández Campos
F. Donelson Smith
Kevin Jeffay
David Ott

SIGMETRICS, June 2001

http://www.cs.unc.edu/Research/dirt

Motivation
Traffic Modeling and Characterization

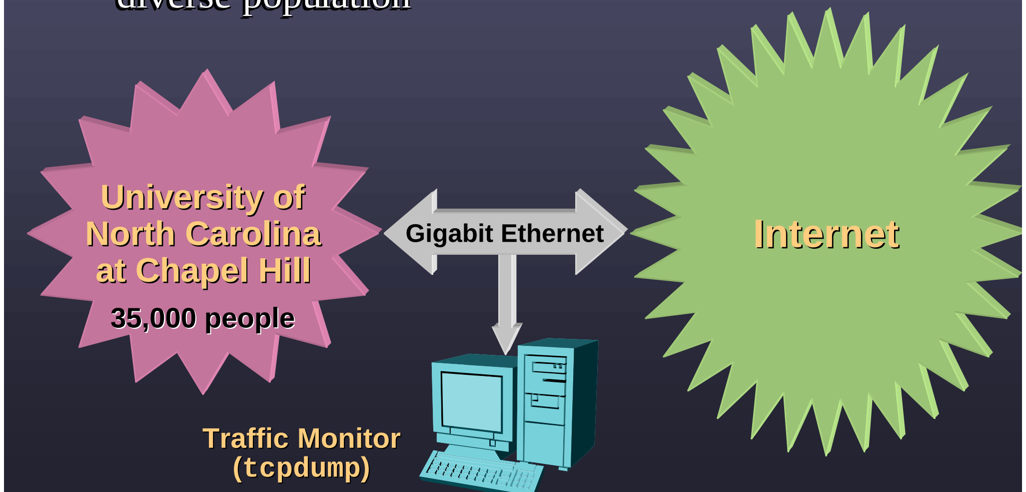
- Can we continuously acquire network traffic data using off-the-shelf hardware and software?
- Can we use this information to construct up-to-date, application-level traffic models?
  - Populate traffic generator with analytic distributions for simulations and lab experiments
- Can we study the traffic generated by a *large population* of users while protecting their privacy?
- Case study: *Web Traffic*

Internet Traffic Characterization
Previous Work

- Traffic modeling before the WWW explosion
  - Danzig et al. (91, 92)
  - Paxson (94)
- Browsing-based web traffic models
  - Mah (95)
  - Crovella et al. (95, 98)
- Models of TCP connections in the web
  - Cleveland et al. (00)
- Other large-scale trace analyses related to the web
  - Gribble & Brewer (97), Balakrishnan et al. (98), Wolman et al. (99), and Feldmann (00)

Methodology
Trace Acquisition

- Study Internet traffic generated by a large and diverse population



Methodology
Benefits of TCP/IP Header Tracing

- Light-weight
  - Off-the-shelf hardware
  - Freely available software
- Privacy
  - Easy to address by anonymizing IP address offline
- Efficient
  - Reduces storage requirements
    - E.g. 161 GB for headers instead of 803 GB for entire packets
  - Reduces processing requirements during tracing
    - Header extraction and recording only
- Large-scale
  - E.g. 7 days x 12 hr, 1 Gbps link (20% avg. util.), 35K users

Trace Collection
Summary

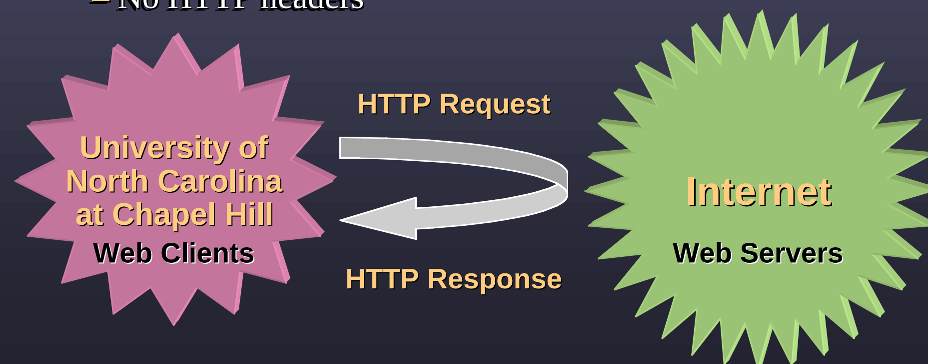
- Three sets of traces from UNC
  - October 99, October 00, April 01
  - 1 hour-long tracing periods (1-6 GB per trace)
  - 42 traces in each set
- Two sets of traces from NLANR (for comparison)
  - October 99, October 00
  - 2 sites
    - San Diego Supercomputing Center
    - Univ. of Michigan/Merit
  - 90 second tracing periods (3-67 MB per trace)
  - 58 traces in each set

Trace Collection
Summary

		99	00	01
Packets	Total	525 M	1873 M	2419 M
	TCP	85%	91%	91%
	HTTP	38%	29%	28%
Bytes	Total	212 GB	721 GB	905 GB
	TCP	86%	90%	91%
	HTTP	56%	35%	36%
Total Traces Size		36 GB	127 GB	164 GB
Avg. % of Packets Lost by Monitor		0 %	0.02 %	0.003 %

Case Study: Web Traffic
Packet Capturing

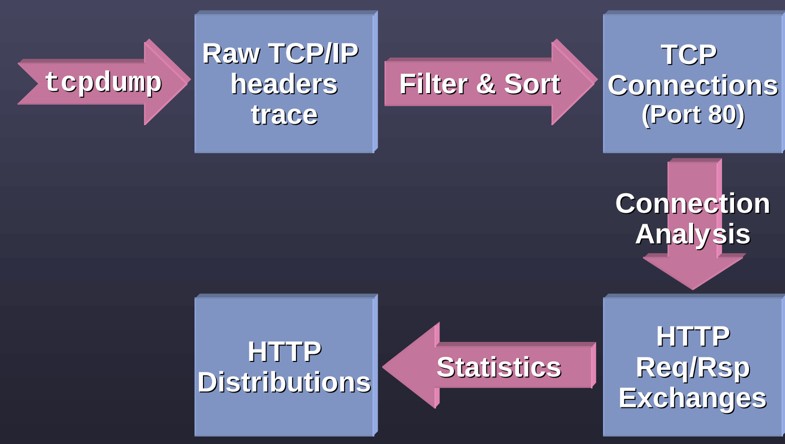
- We study a large collection of users as web content consumers
- We only capture TCP/IP headers
  - No HTTP headers



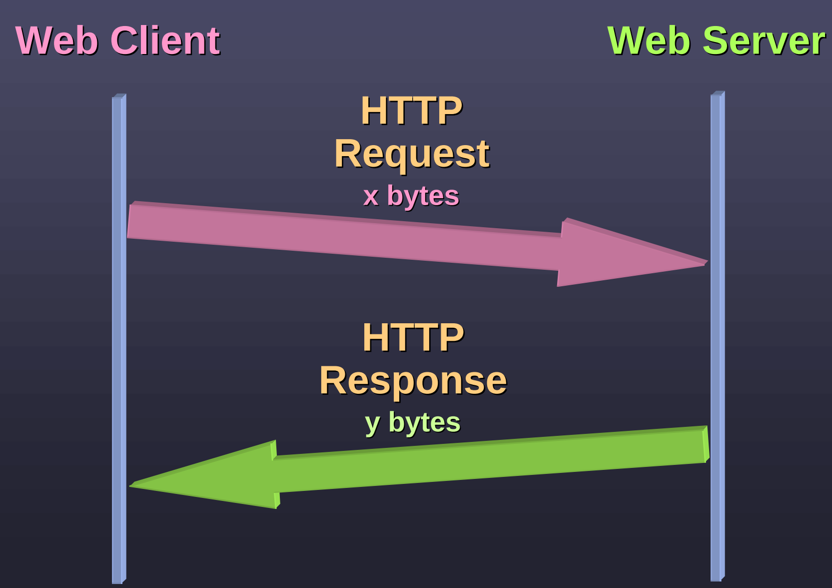
Methodology
Do We Really Need HTTP Headers?

- We can infer plenty of HTTP information from TCP/IP headers
  - Request size
  - Response size
  - Embedded objects per web page
  - Servers per page
  - Use of persistent connections
  - ...
- TCP/IP headers are sufficient for
  - Constructing application-level traffic models
  - Studying the impact of new HTTP dynamics

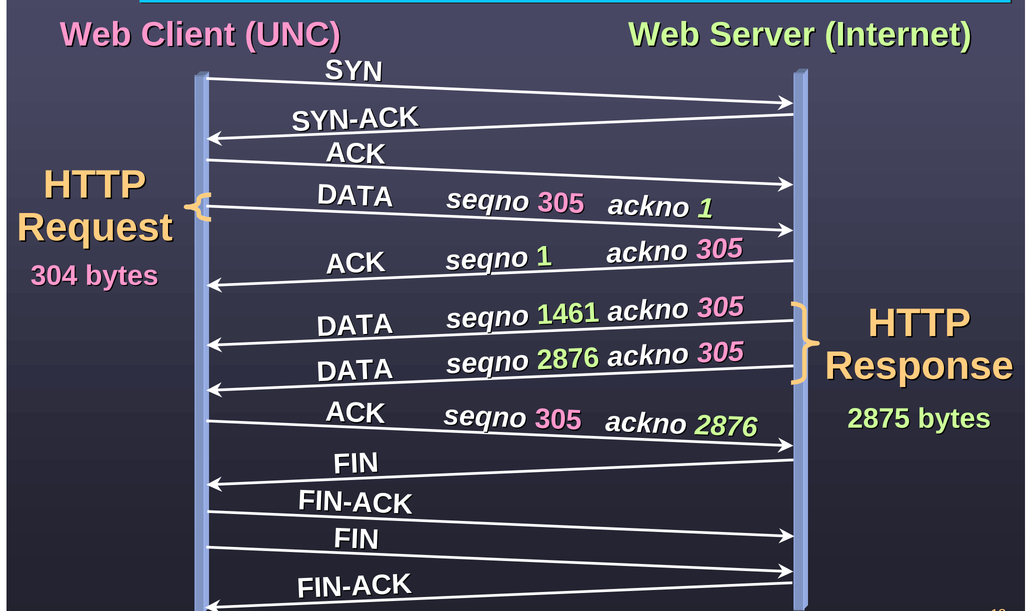
Web Traffic Analysis
Processing Sequence Overview



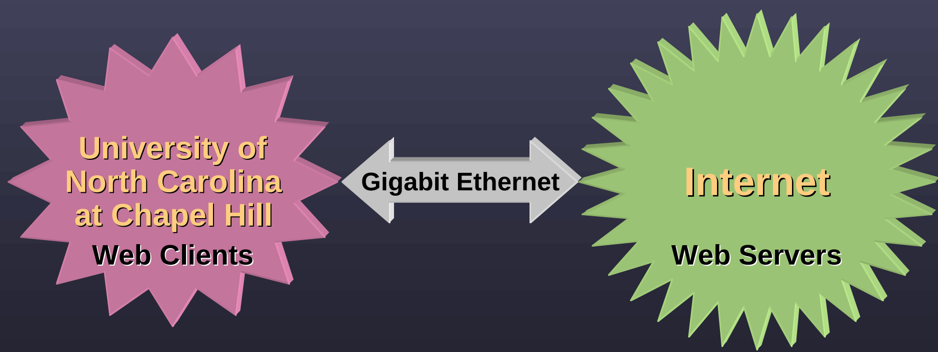
Methodology
Request/Response Traces



TCP/IP Headers and HTTP
Request/response Exchange

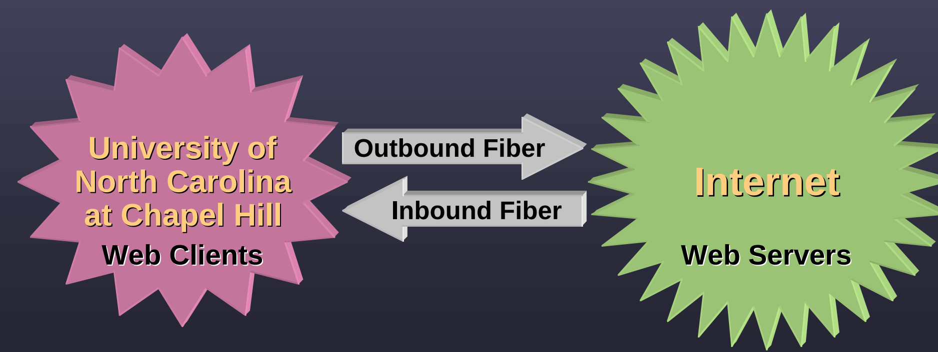


Packet Capturing
Inbound TCP/IP Headers Only



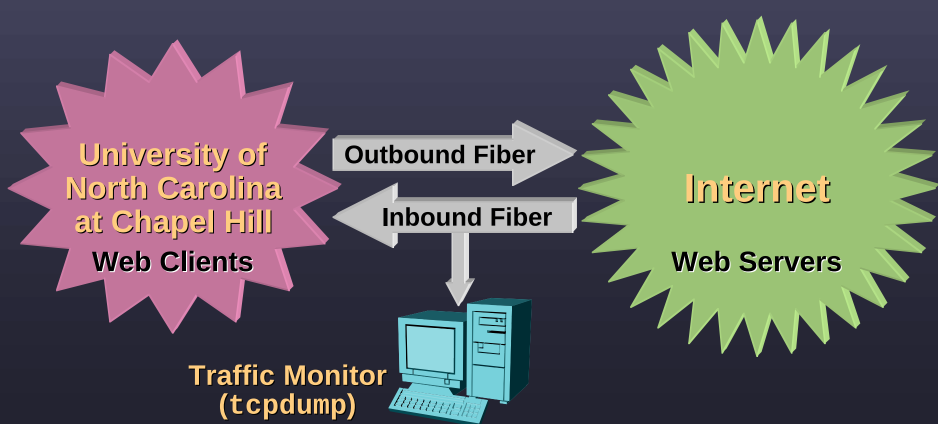
Packet Capturing
Inbound TCP/IP Headers Only

- Two fiber links

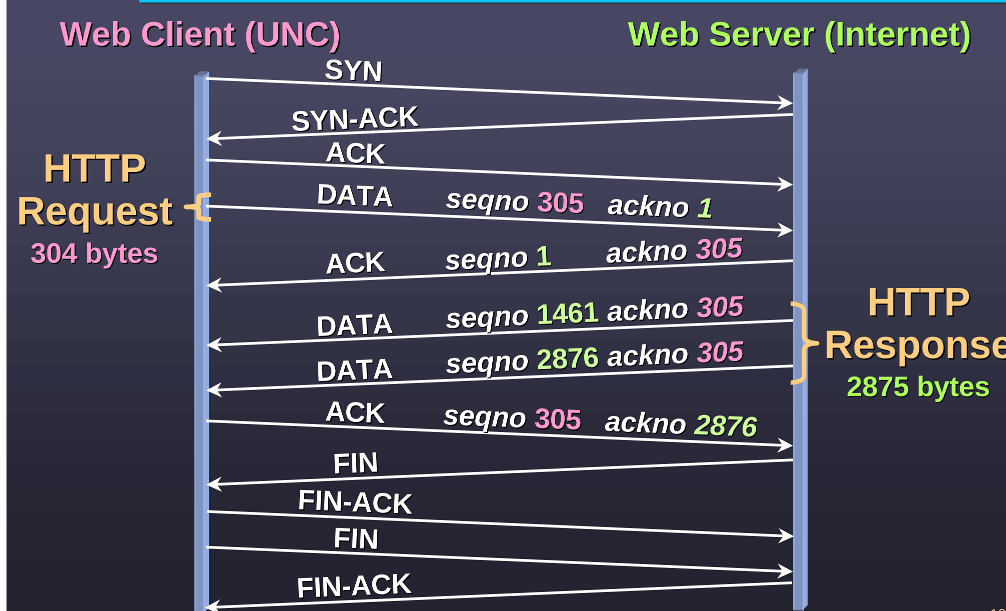


Packet Capturing
Inbound TCP/IP Headers Only

- Only inbound TCP/IP headers are captured
  - Eliminate synchronization and buffering issues on the NIC
  - Reduce trace size

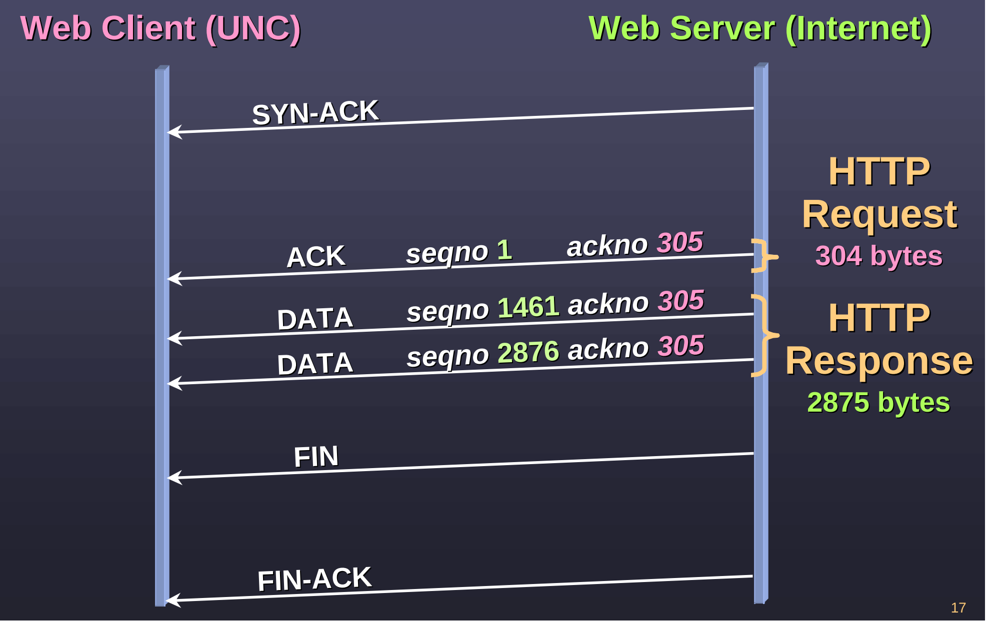


TCP/IP Headers and HTTP
Request/response Exchange



TCP/IP Headers and HTTP

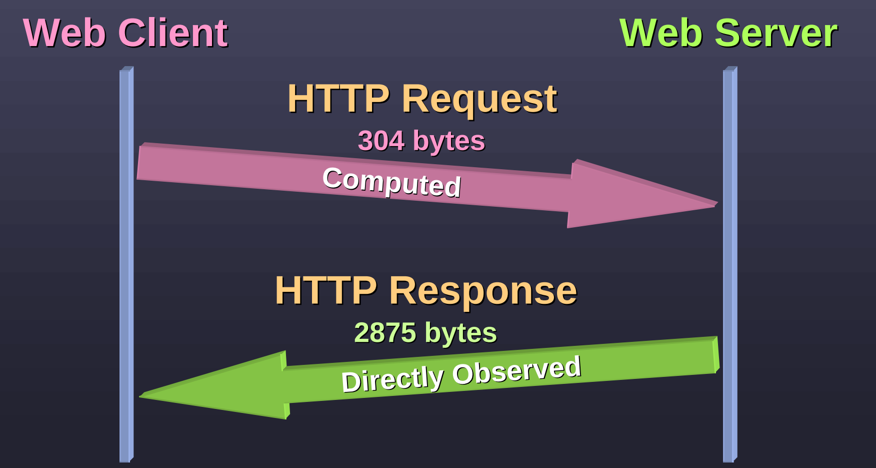
Server-to-client Segments Only



Methodology

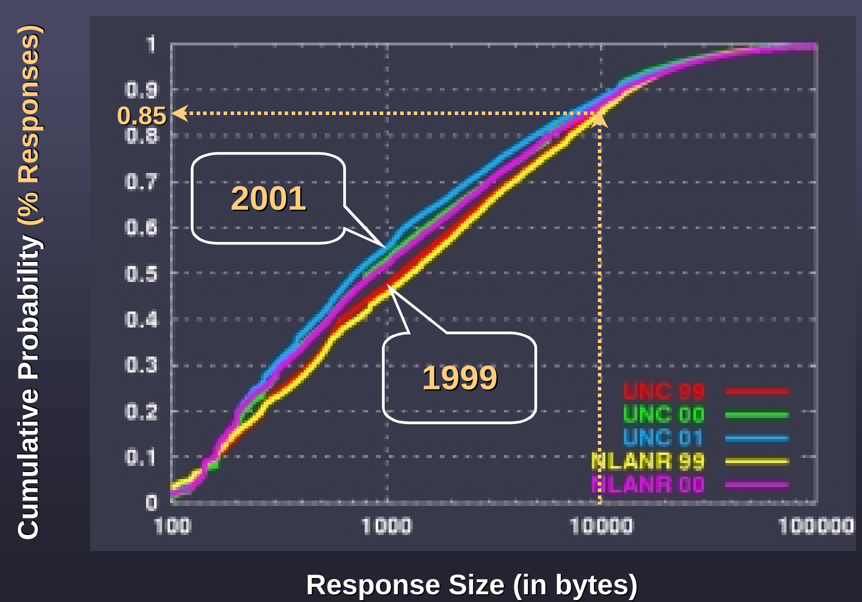
Request/Response Traces

- Unidirectional TCP/IP header traces are sufficient for capturing application-level behavior



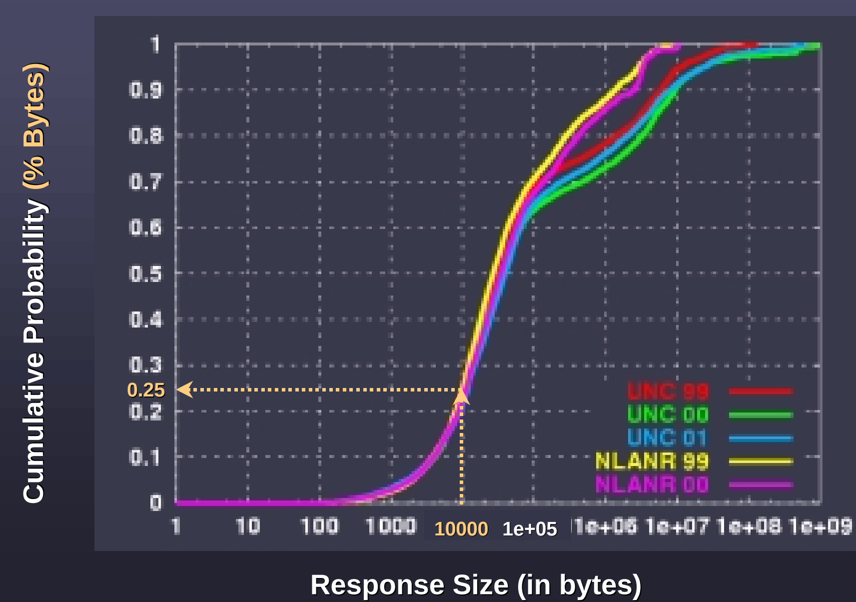
HTTP Characterization

Response Data Sizes – Body CDF



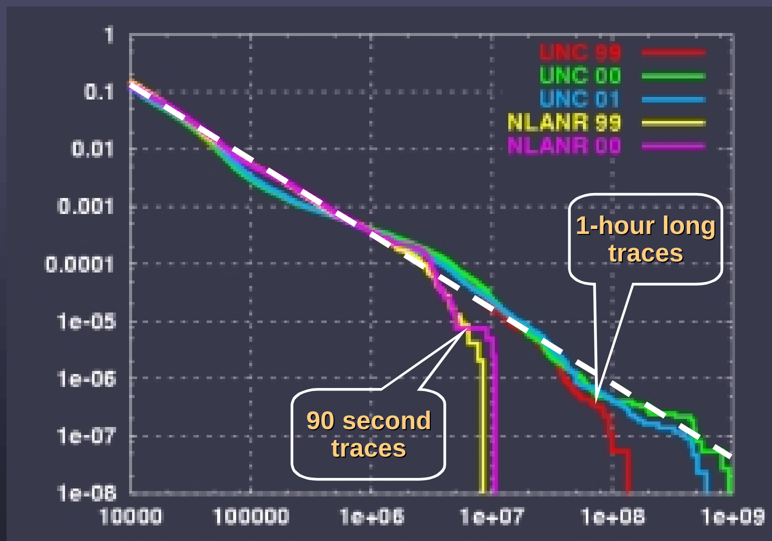
HTTP Characterization

Response Data Volumes – Body CDF



HTTP Characterization
Response Data Sizes – Tail CCDF

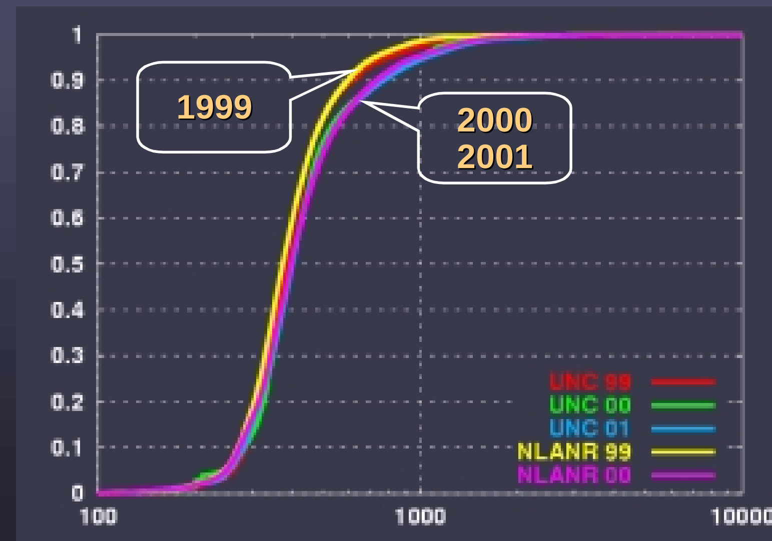
Complementary Cumulative Probability



Response Size (in bytes)

HTTP Characterization
Request Data Size – Body CDF

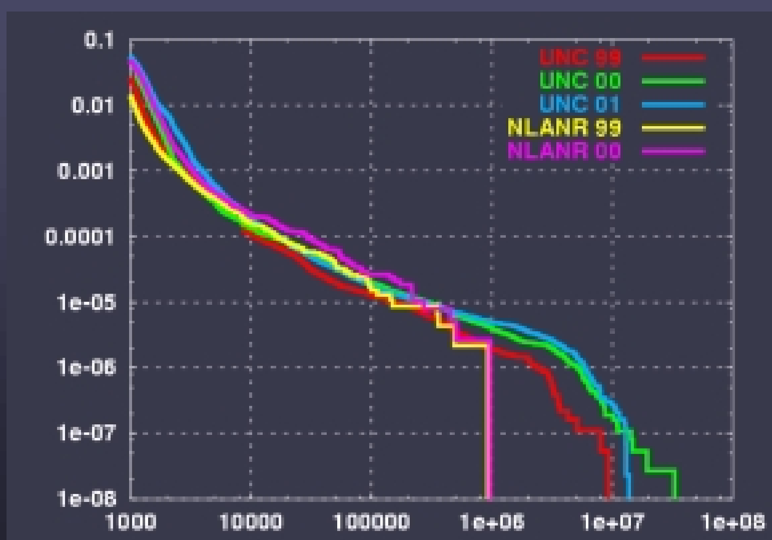
Cumulative Probability



Request Size (in bytes)

HTTP Characterization
Request Data Size – Tail CCDF

Complementary Cumulative Probability

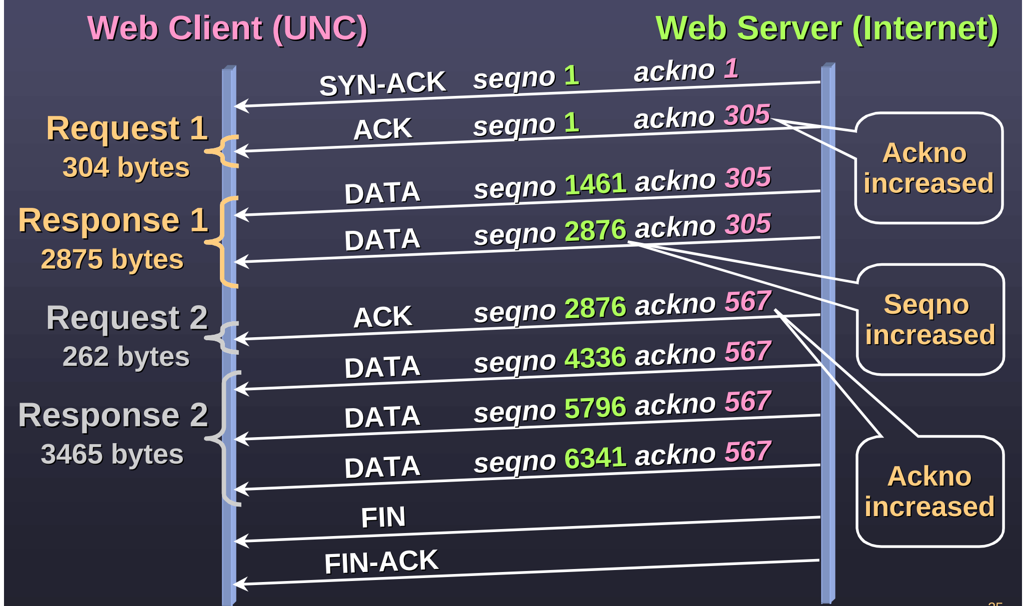


Request Size (in bytes)

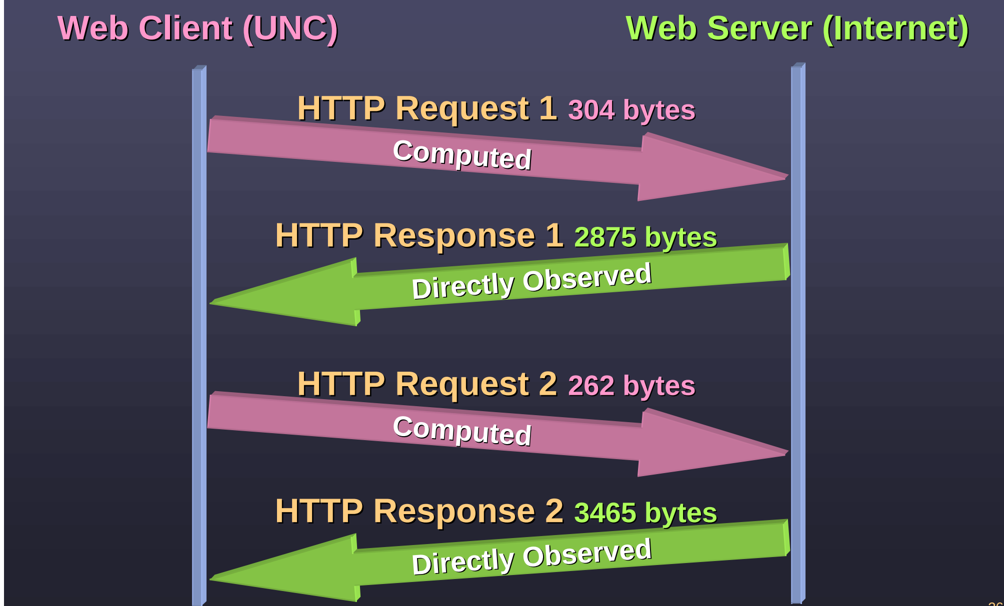
Persistent Connections in HTTP
Effective Persistence

- An HTTP *persistent connection* can use a single TCP connection to carry one or more request/response exchanges
- This feature is supported in newer versions of the protocol
  - HTTP/1.0 (limited support)
  - HTTP/1.1
- We study how persistent connections are used
  - We define *effective persistence* as *two or more* request/response exchanges in the same TCP connection

Persistent Connections in HTTP
Example – TCP/IP Headers



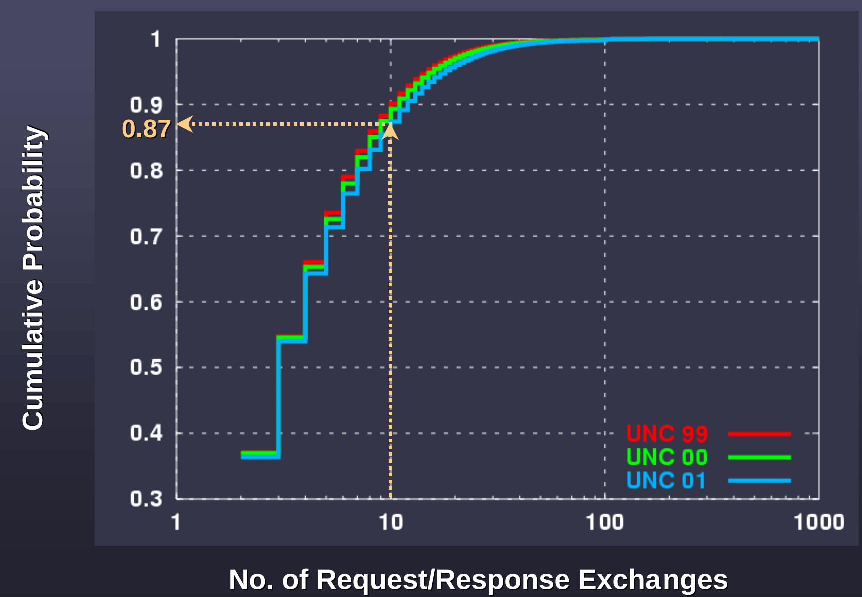
Persistent Connections in HTTP
Example – Request/Response Exchanges



Effective Persistent Connections
Summary Statistics

		UNC 00	NLANR 00
Connections	Non-Persistent	78.1%	63.4%
	Persistent	15.1%	13.8%
	Unclassified	6.8%	22.8%
Objects	Non-Persistent	50.3%	57.2%
	Persistent	49.7%	42.8%
Bytes	Non-Persistent	49.6%	54.3%
	Persistent	40.4%	35.7%

HTTP Characterization
Objects in Persistent Connections



HTTP Characterization
Other Statistics

- Page-based statistics (based on Mah and Crovella et al.)
  - Think times
  - Top-level vs. embedded objects
    - » *Requests and Responses*
  - Unique TCP connections per page
  - Unique server IP addresses per page
  - Consecutive pages per server
  - Number of pages per client
  - Primary vs. secondary servers
    - » *Requests and Responses*
- Other non-page-based statistics
  - Number of exchanges per client

Limitations
TCP/IP Header Tracing

- *Uncertainties* arise when application-level information is inferred from transport-level headers
- We discuss several issues in our paper
  - Pipelining
  - User/browser interactions
    - » *Stop and reload*
  - Caches
    - » *Local cache and proxies*
  - TCP segment processing
    - » *Segment reordering*
- In summary, limited or no impact in our results

Summary and Conclusions
Methodology

- Unidirectional TCP/IP header tracing is a powerful and light-weight traffic measurement methodology
- Limitations have a minor impact in application-level results
- We also applied this methodology to
  - SMTP
  - FTP
  - Other application-level protocol

Summary and Conclusions
Web Traffic Characterization

- New data to populate traffic generators
  - Request sizes
  - Response sizes
  - Use of persistent connections
  - ...
- 1-hour long traces are sufficient to capture application-level behavior
  - Short traces cut off large objects, which skews the tails of the distributions
- Persistent Connections:
  - ~15% of all the HTTP connections
  - 40-50% of all the transferred HTTP bytes