A Protocol-Independent Technique for Eliminating Redundant Network Traffic

Neil Spring and David Wetherall
University of Washington

SIGCOMM
August 30, 2000

Motivation

Web caching misses repetition

- Database queries, web searches, personalized pages
- Proxy cache hit rate may peak at 50%

Is there repetition in uncached content?

Can we use it to reduce bandwidth demands?

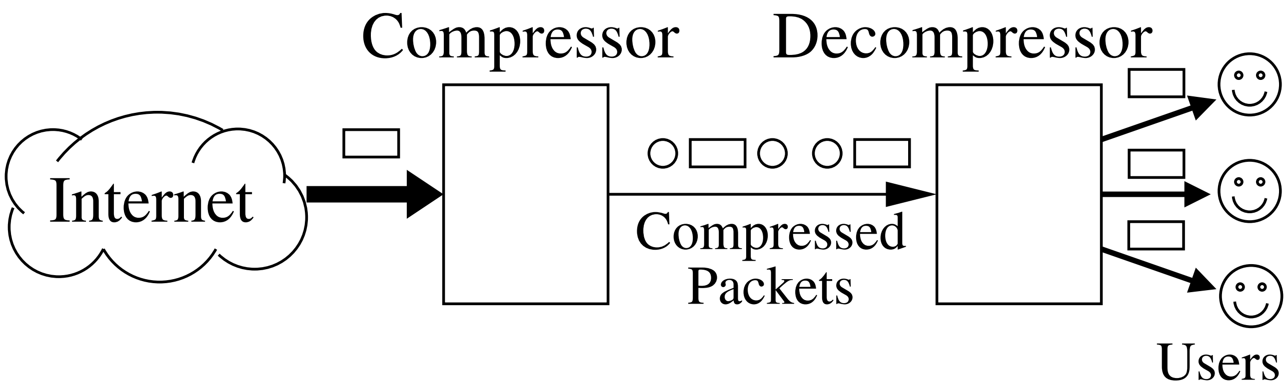
Teaser:

- 40% byte savings in uncacheable documents
- 4-34% byte savings in other protocols

Finding repetition

Evaluate a compressed tunnel:

- tunnel endpoints implement compression
- tunnel traverses a slow or expensive link



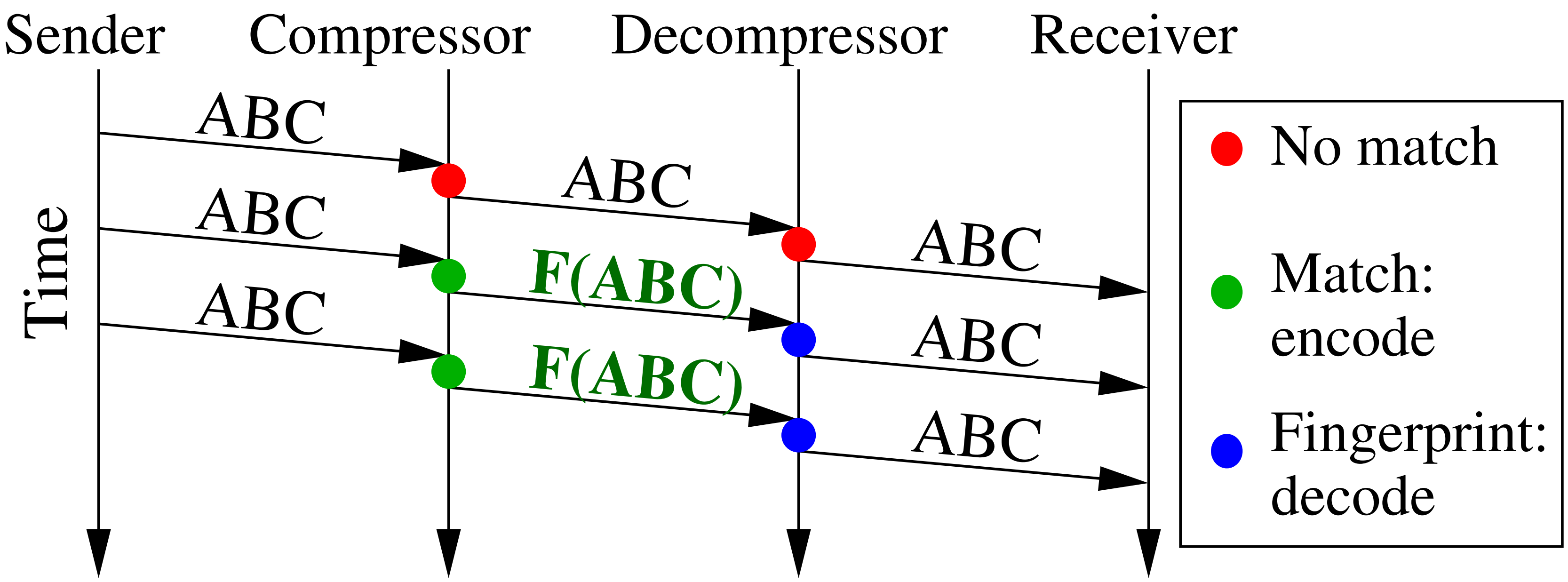
Compressor: detects and encodes repetition

Decompressor: decodes using a shared dictionary

Background: Whole packet caching

Our compression builds on [Santos & Wetherall]:

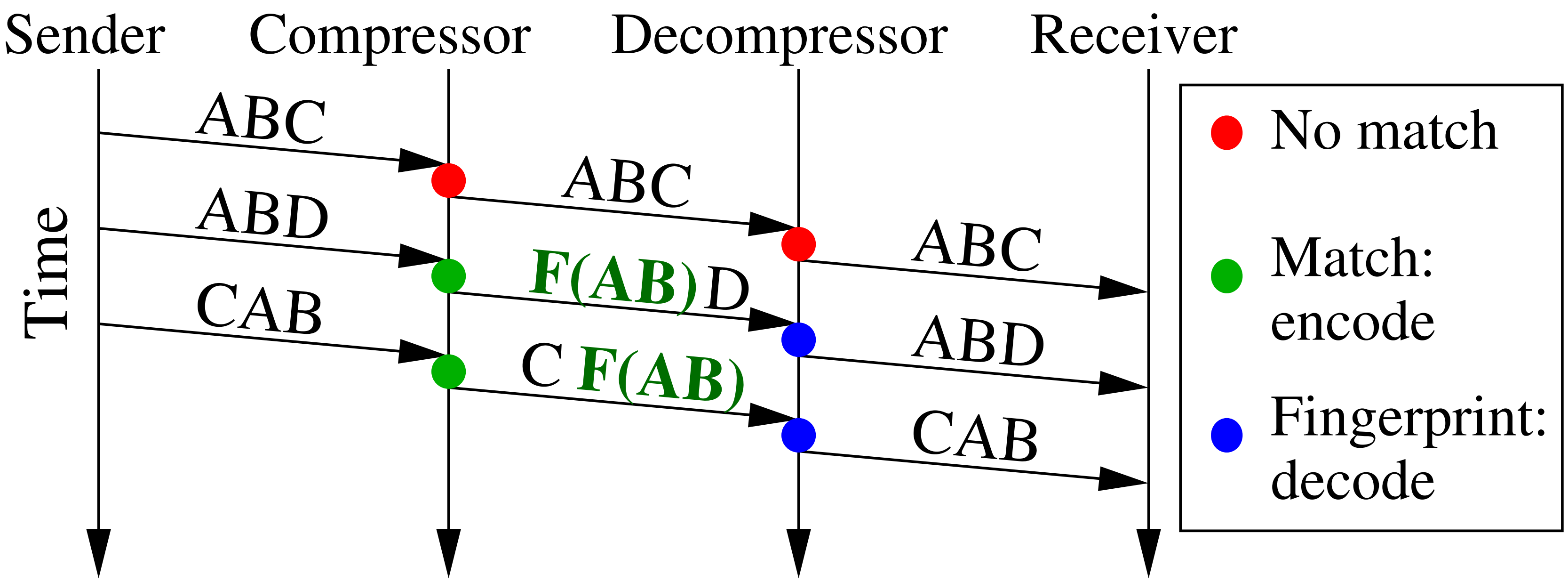
- If fingerprint of packet payload is in index
  - send fingerprint: symmetric cache decodes



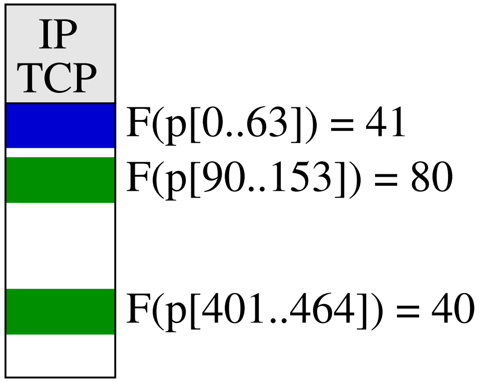
Redundancy Suppression

Fingerprint 64-byte substrings of payload

Send fingerprint in place of substring.



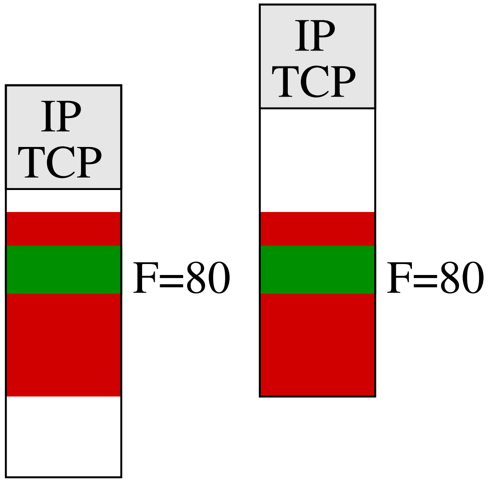
Efficient Redundancy Suppression



Calculate all, but index only fingerprints:

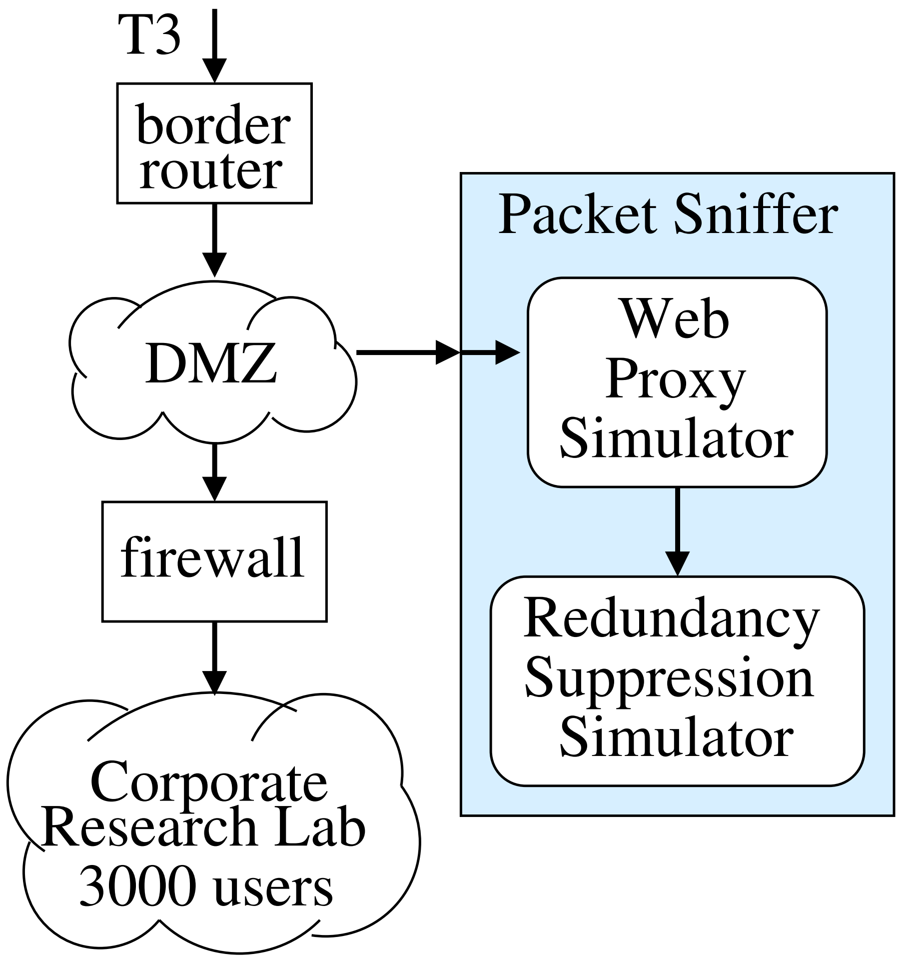
- that begin a packet
- that end with 5 binary zeroes

Inspired by Manber's tool: siff



On a match, look up stored packet,
Search for additional repeated bytes ( ) before and after fingerprint match

Experimental Setup



Web proxy simulator: filters out "cached" traffic

576-byte MTU

100MB History Window

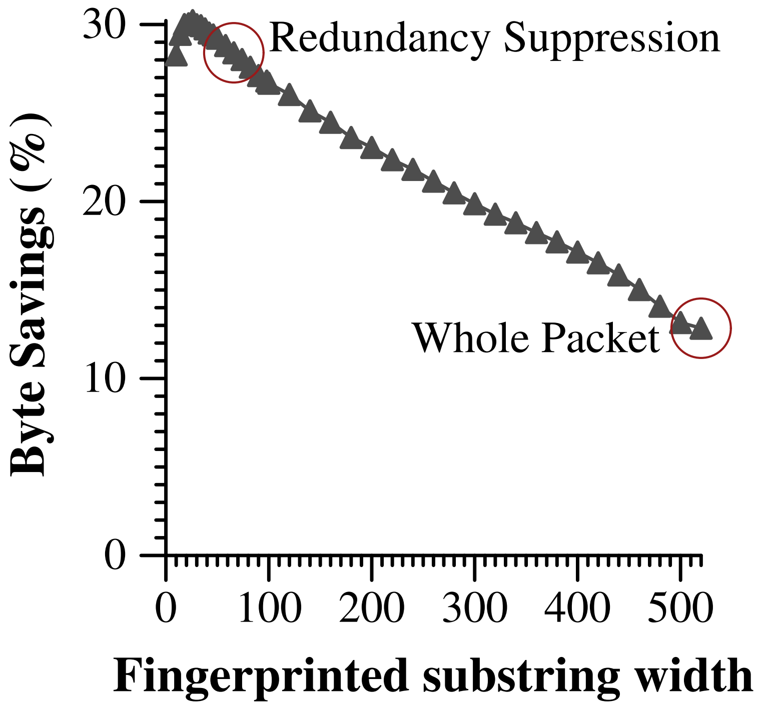
12-byte per match encoding penalty

Byte savings is repetition minus encoding penalty

Redundancy Suppression vs. Whole Packet Caching

Basic packet caching:

Fingerprinted substring width = packet size



Off-line trace to find best fingerprint width

2x improvement by matching substrings

Protocols

Filter by well-known port

When repetition is found, tally it for that protocol

Ranges from 4-34% byte savings

Notes (full breakdown in paper):

- Web: 64.3% traffic, 30% savings
- Telnet: 0.1% traffic, 34% savings
- Napster: 2.7% traffic, 6% savings

Most traffic and significant benefit in Web.

Web caches and repetition

Web requests and responses tagged

cacheable, uncacheable & why

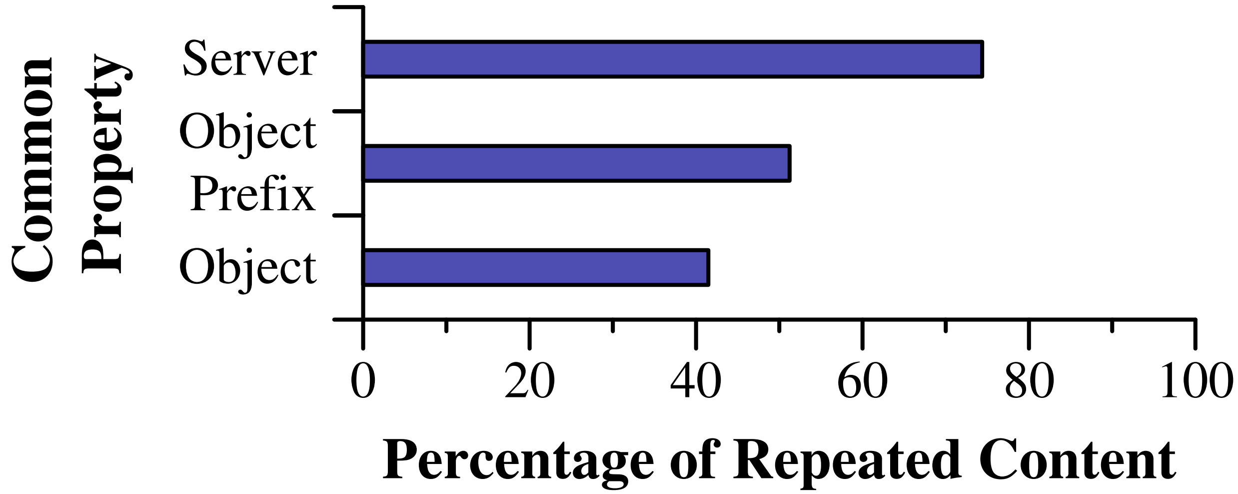
Cold misses: 15.6% repetition

Uncacheable documents: 41.4% repetition

- Highest repetition (76%) in “pragma response” class.
- Lowest repetition (3.3%) in “push” and (17.4%) in “method” class.

Source locality

For HTTP connections, when repetition is found,
Which characteristics do the connections share?



Bias:

- Server IP address / DNS round robin

Conclusions

There’s lots of repetition!

Significant repetition in uncacheable Web documents

Most (80%) repetition between Web documents from the same server:

Web-specific compression across all documents from the same server would be effective.

gzip, zlib, and Lempel-Ziv 77

Feature	**zlib**	**Redundancy Suppression**
Compresses	streams	packet payloads
History Window	32KB	100MB +
Fingerprints	15b of 4B	60b of 64B
References	offset	fingerprint
Indexing	dense	probabilistic
Runs	after	before and after
Byte Savings	14-36%	54%