Lecture 9/10: Feb 23.

1. Last class: motivation for RE service
2. RE algorithms (MAXP and MODP for identifying redundancy)
3. Empirical study and cache size
4. Cache size fast and effective RE
   - avoid duplicate storage
   - avoid expensive encoding at every hop
   - decouple decoding from its immediate reliance on encoding.

Network-wide optimization using a central routing platform

- Collect traffic and redundancy properties
- Router computation & memory capacities (only for decoding)
- Compute fraction of traffic on a path that a router should cache such that
  - cache occupancy is minimized
  - lookups per second match & router constraint
  - overall network footprint is minimized

- Translation solution in encoding and decoding
  - hash ranges that are non-overlapping
  - bucketize each router's hash so that packet eviction only happens in a bucket corresponding to a router SG pair
- look at overlap of paths to know if a packet is encodable.

3Gud — 2.5 Gbps for encoding
12 Gbps for decoding.

Applications — partial/incremental deployment:
- data-centers (serves can also participate)
- multi-hop wireless

Let's now look at the redundancy-awareness paper.

- Implications of universal RE on network protocols.
- Initial ideas on fast stand-alone RE implementation

Impact on
RE-awareness impacts both intra and inter-domain routing.

Both cases: estimate redundancy profiles
compute routes that maximize potential for compression.

Intra-domain case: egress selection
                  peer selection

Redundancy awareness: 1) for the same peer, pick exit point
                      in redundancy-aware manner than
                      based on latency. (compare redundancy with
                      minimally delayed traffic)

Allows you to combat load due to peer's traffic
2. Keep egress point selection fixed, but push peers in a redundancy-aware manner than based on BGP policies. Trade-off external and internal costs could be done on demand.

3. Adapt both egress and peer selection to redundancy very fine-grained trade-offs such as the above.

4. Greater benefit if neighbors coordinate in their redundancy-aware route selection.

Network vs. end-host RE: Concept: enterprise-DC link or even more generally speaking.

Downsides of network-based approaches:

- Encryption
- No last-hop benefits

Why not push as high up into end-hosts as possible:

- Socket layer as it avoids the need for reinventing RE.

Question: is there enough redundancy? Our measurements say so.

Challenges: optimal use of memory and storage on servers and clients.

Take a look at RE algorithms and data structure again.
Server

PP store
Content store

Client

PP store
Content store

$S \rightarrow$ size of packet, $W \rightarrow$ window, $P \rightarrow$ "sampling rate"

\[
\text{FIXED} \quad \text{MOD P} \quad \text{MAX P} \quad \text{SB}
\]

Markers:

- Every $p^{th}$ byte
- Hash of $w$-byte region around marker
- Rolling R-K hash of $w$-byte region around marker

\[
\text{cost} = H\left(\frac{S}{p}\right) + HK(w) + H(S-w) + A(s) + H\left(\frac{S}{p}\right) + A'(s) + H\left(\frac{S}{p}\right)
\]

Efficiency rank:

\[
A \quad a \quad 1 \quad 3
\]

chunk
Match

\[
\text{compute} \quad (SHA-1)
\]

Hashes of byte regions between markers.

Client

chunk hashes
content

chunk hashes

Client does not need chunk hash store. Server can simply send exact client memory address and emulate client eviction policies.
Server does not need to store content as content is stored at the client. Servers only need to do LRU eviction in cache store and assume client does similar eviction in his content cache. Control client cache eviction and eviction behavior.