Domain Name System as a Memory and Communication Medium

Dušan Bernát
(bernat@fiit.stuba.sk)

Institute of Computer Systems and Networks,
Faculty of Informatics and Information Technology,
STU Bratislava, Slovakia

Overview

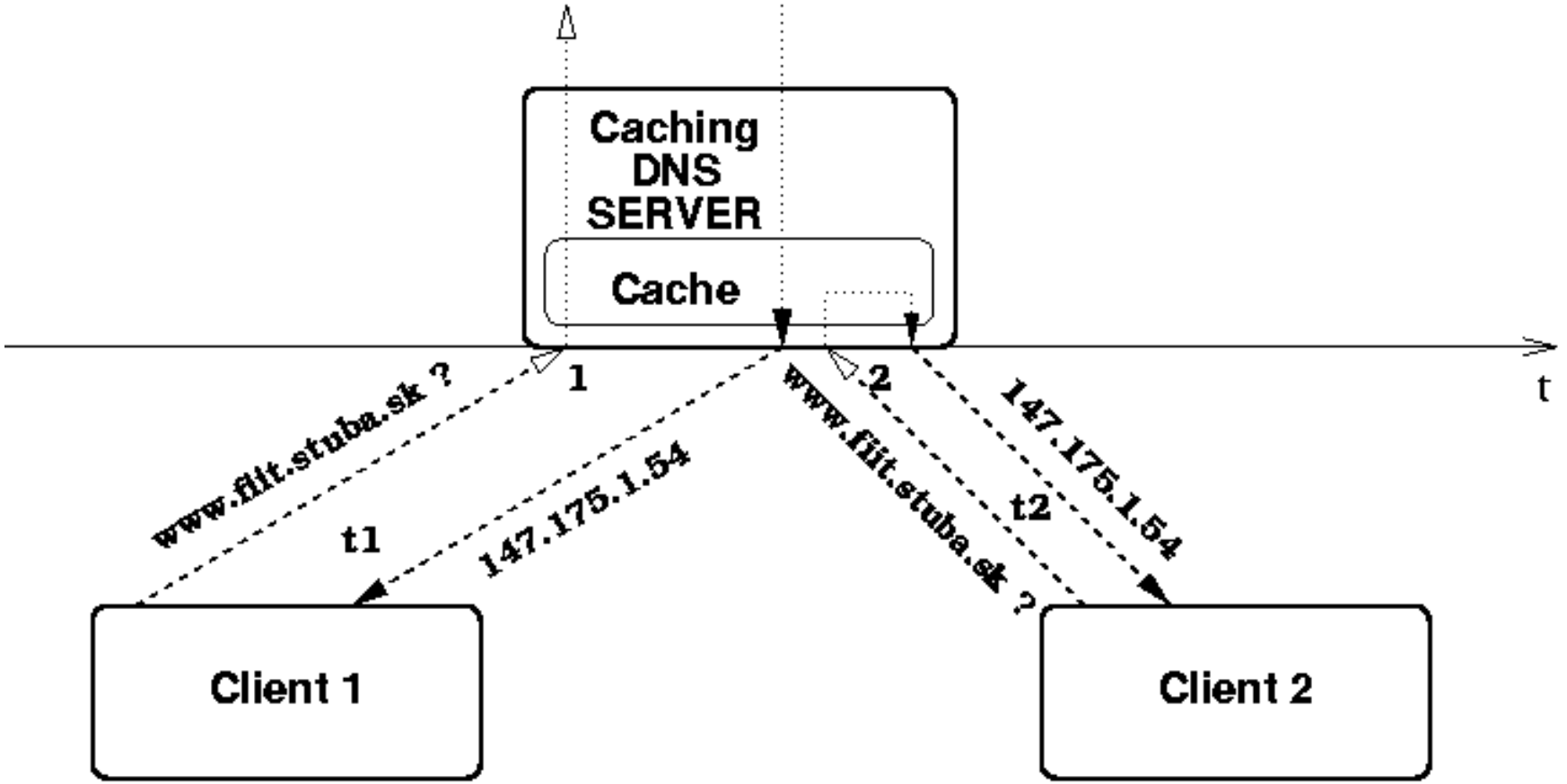
- **Communication** – in an unusual way, indirect, hidden, ...
- **Medium** – DNS, cache, ...
- **Protocol** – principle (time based), implementation, properties, ...
- **Some results** – error rate, optimal performance, purpose, ...
- **Future work** – synchronisation, error correction, adaptability, ...

Domain Name System

- Translates symbolic names into IP addresses (among other things).
- `www.fiit.stuba.sk has address 147.175.1.54`
- Extensible, hierarchical, distributed, ...
- Several DNS servers may be involved in single name resolution.
- Caching name servers reduce response times.

DNS Cache

- Consecutive queries for the same name



Storing one bit

- Result of caching:
  - $T_{12} < T_{TTL} \Rightarrow t_2 \cdot M < t_1$ for some $M \geq 1$
- T_{12} is interval between two consecutive queries for the same name.
- T_{TTL} is Time To Live of cache record.
- Measuring the response **time** we can decide whether the bit is set or not.

Building a memory

- Write operation:
  - **`write(addr,1)`**: send query for **`addr`**,
  - **`write(addr,0)`**: do nothing.
- Read operation:
  - **`read(addr)`**: send two consecutive queries for **`addr`**, measure and compare response times.
  - If the two response times have similar values (both are short), the bit has been set,
  - else (the first one is greater) it is not set.

Address space

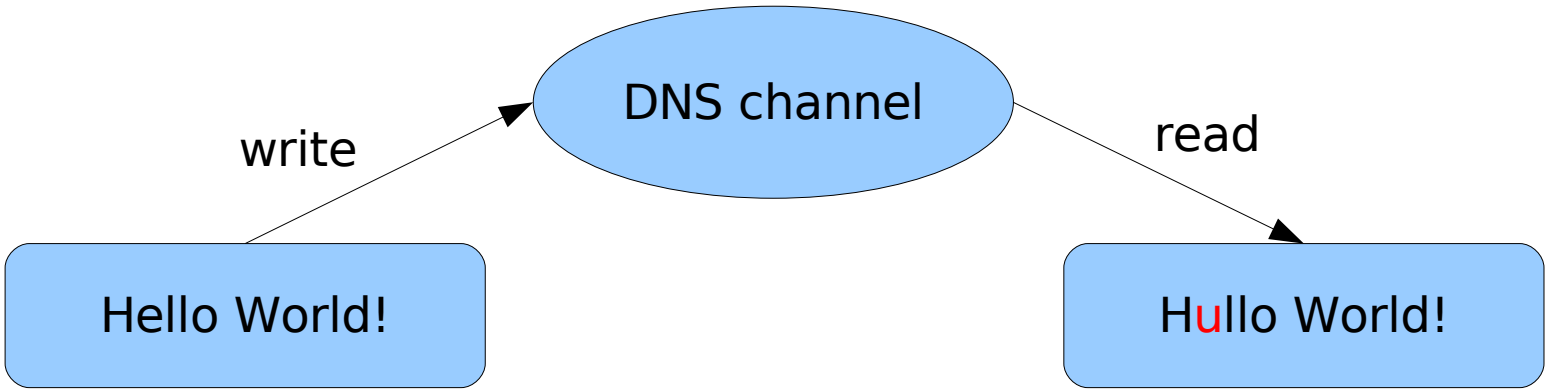
- Ordered subset (sequence) of domain names.
- Non-existent addresses
  - does not conflict with regular traffic,
  - negative caching (since RFC 2308).
- Can be spanned over several DNS servers
  - may reduce possibility of detection.
- It must be know prior to communication to both parties.

Memory properties

- High capacity
  - address space is formed by all legitimate DNS names, depends on actual cache size.
- Access time
  - read time depends on the values read,
  - it can be increased on the expense of visibility.
- Read-once behaviour
  - data destroyed during read.

Properties – Example

- Error rate
  - the response time may depend on many random influences and network delays.
- Sending a message ...
  - well, it is possible :-)



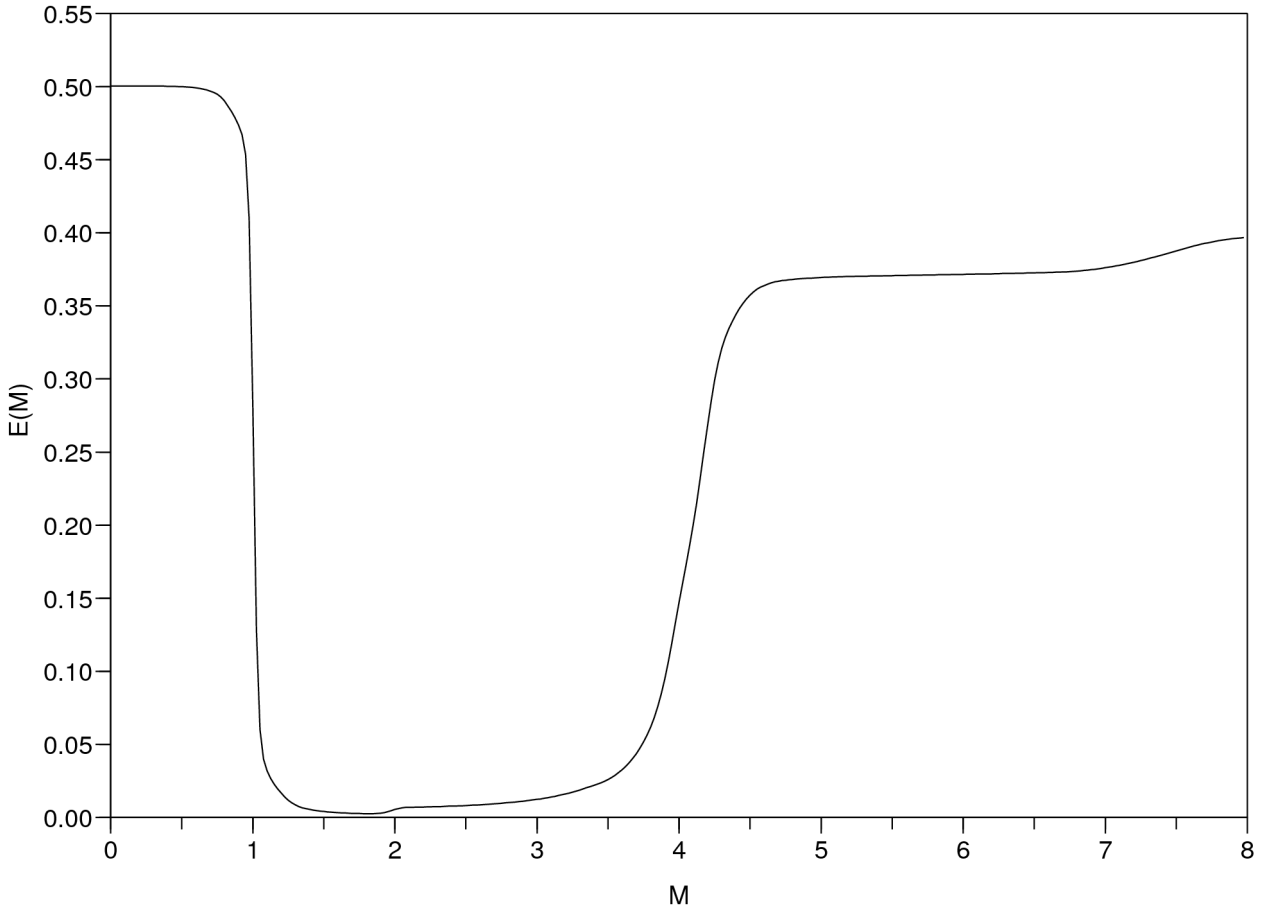
Read operation

- Assume we have measured two response times (t_1, t_2):
  - What is the result of read operation?
  - $B_1(M, t_1, t_2) \equiv \neg(t_2 \cdot M < t_1)$
  - Relation B_1 is true (and evaluates to 1) if measured values corresponds to reading 1.
  - This is the case when both queries are served from the cache so both response times are roughly equal (t_2 is not M times shorter).

The M

- What role does play factor M?
  - How large is the difference in response time?
  - If the second one is M times shorter than the first one we get 0 (record was not cached).
  - Thus all values read from the memory depend on M.
- What is the proper value of M?
  - It depends ...
  - But we want to minimise $E(M) = \frac{|D, D_r(M)|_H}{N}$

Error rate dependence on M



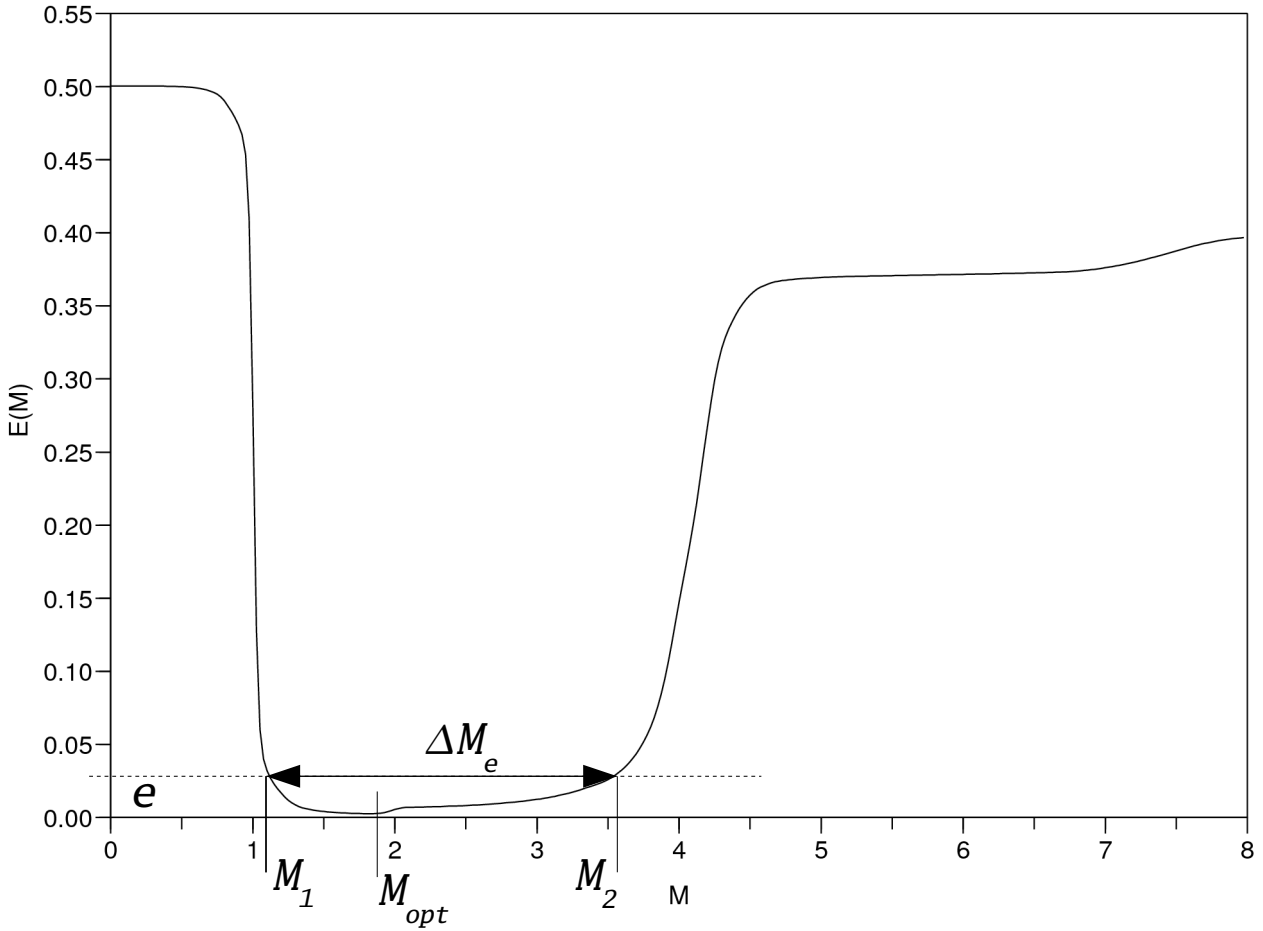
Unusable values

- The first (unusable) region $0 < M < 1$
  - $M < 1 \wedge B_1(M, t_{1,} t_2) \Rightarrow t_1 < t_2$
  - This is only due to random delays in the net.
  - For all practical cases we have:
  - $M < 1 \Rightarrow B_1(M, t_{1,} t_2) \rightarrow 0, \forall (t_{1,} t_2)$
  - If we assume a block of random uniformly distributed bits, we get $E \rightarrow 0.5$
  - Error rate is constantly at maximum value.

Useful values

- As we move slightly to the right from *1*, error rate rapidly falls down.
  - *M=1*, phase transition.
- Here is where the transfer is possible.
  - The best possible performance for:
  - $M_{opt}: E(M_{opt}) = min\{E(M); M > 1\}$
  - In practice, memory works properly for values of M from interval:
  - $\Delta M_e = M_2 - M_{1,} E(M_1) = E(M_2) = e \wedge M_2 > M_1$

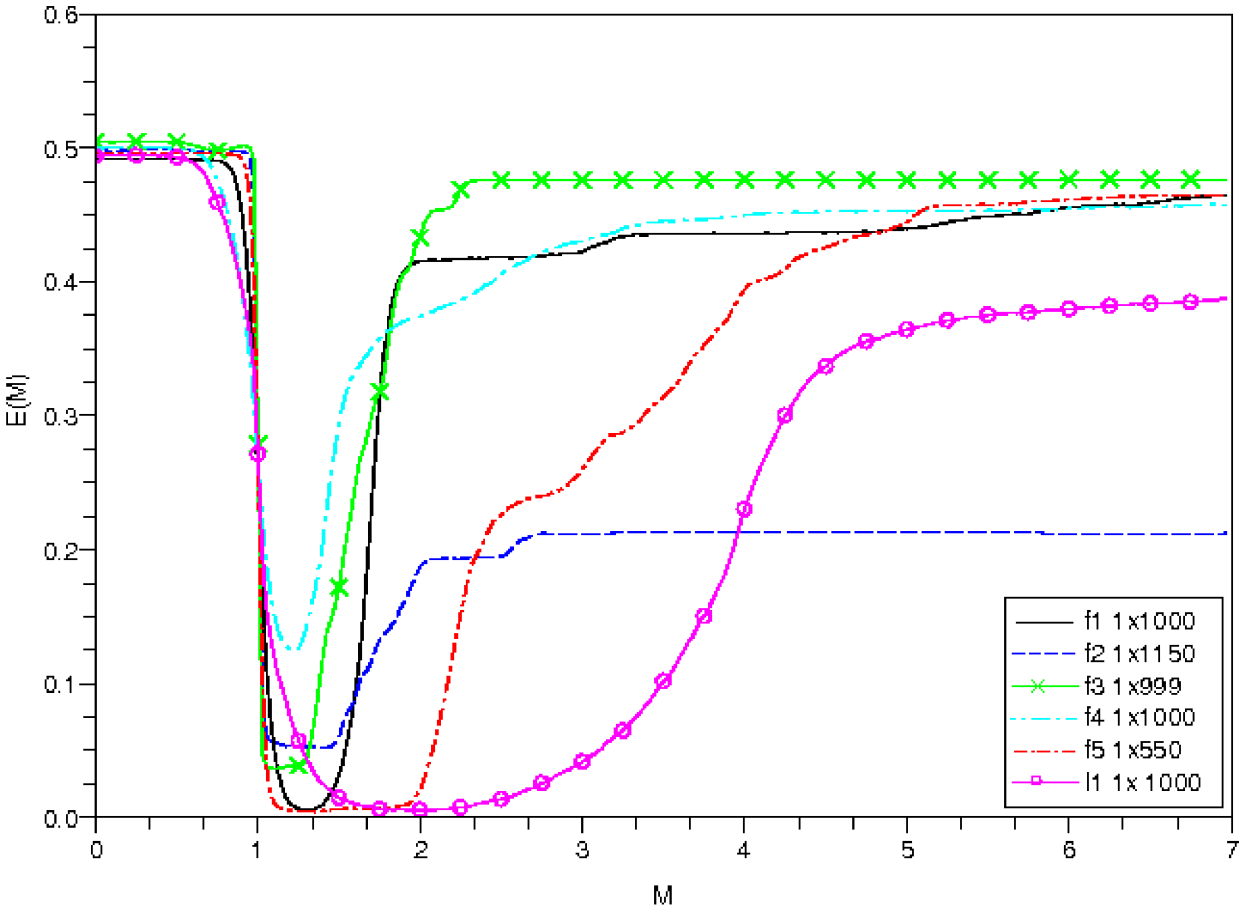
Limits of the error rate



... and bad again

- Third region
- For $M \gg M_2$ error rate tends to 0.5 again.
  - $M \gg M_2 \Rightarrow B_1(M, t_1, t_2) \rightarrow 1, \forall (t_1, t_2) \Rightarrow E(M) \rightarrow 0.5$
  - The value of M_2 is a characteristic of particular DNS server and network settings.

DNS Fingerprint



Conclusions

- Time based communication via DNS cache is possible.
- It can be improved by:
  - using error correcting codes,
    - (it is straightforward, but decreases speed)
  - adding synchronisation mechanism to provide usual synchronous/asynchronous read and write operations,
    - (done, thought not shown here)
  - self-adjustment of M during communication.

Conclusions

- Understanding its mechanism we can prevent its usage:
  - not allow anyone to use DNS server,
  - delay and/or reorder consecutive DNS responses (e.g. on a firewall).
- Time based protocol allows
  - to find out some information about the network topology and settings,
  - to make DNS fingerprints.

Thank you…