Analyzing Cacheable Traffic in ISP Access Networks for Micro CDN applications via Content-Centric Networking

Claudio Imbrenda
Luca Muscariello
Orange Labs
Dario Rossi
Telecom ParisTech

Outline

- Motivation
- Monitoring
- Collected statistics
- Trace driven simulation
- Traffic characterization
- Conclusions

Outline

- **Motivation**
- Monitoring
- Collected statistics
- Trace driven simulation
- Traffic characterization
- Conclusions

Motivation

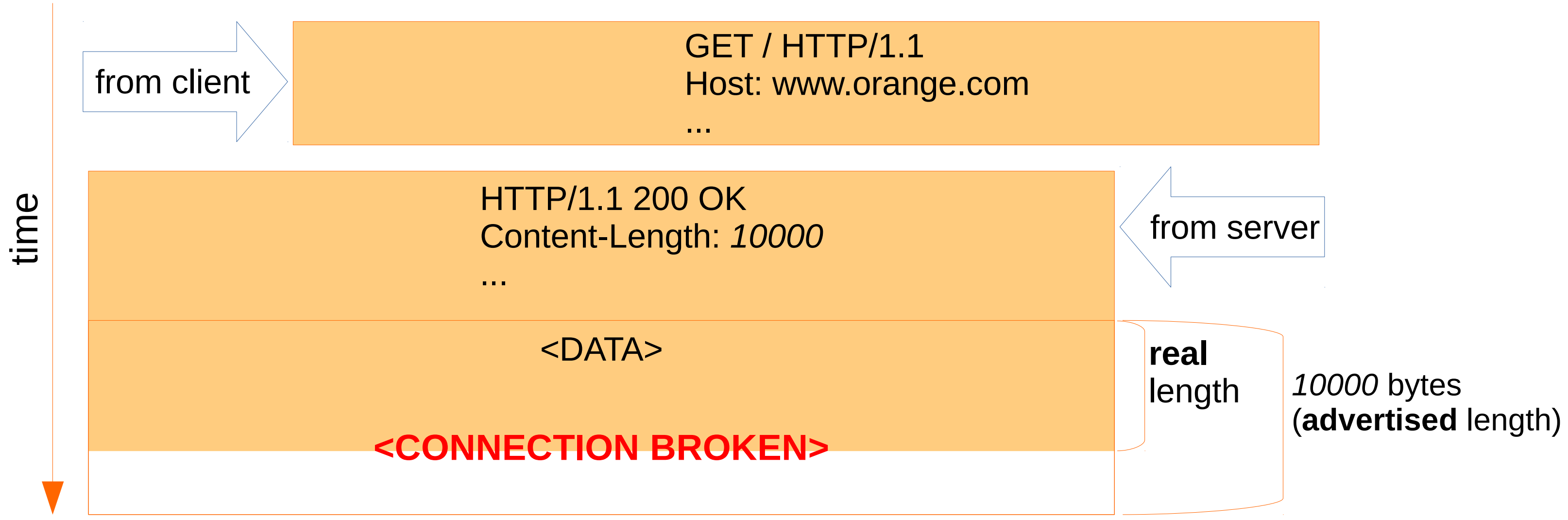
- Traffic is skyrocketing: Web traffic, video streaming
  - From outside of the ISP network
  - Unpredictable
  - Increased investment in access = increased pressure on backhaul
- μCDN: caching at the very edge of the network
  - Caching is a relatively new tool for traffic engineering for ISPs
- Caching has multiple benefits
  - Better quality of service (latency)
  - Lower costs for operators
- Most ICN proposals include caching as a primitive
  - This study therefore helps to validate that approach
- Traffic modeling is also useful

Outline

- Motivation
- **Monitoring**
- Collected statistics
- Trace driven simulation
- Traffic characterization
- Conclusions

HTTP analysis

- A correct analysis of HTTP traffic for caching purposes requires an accurate analysis of the HTTP transactions
  - Not just requests and replies with timestamps, relate the replies to their requests
  - Size of the objects (both *advertised* and *effective*), presence of *ETAG* and cookies
    - We need to identify the objects and their the sizes to estimate the size of the caches



Existing tools inadequate

- A tool has been developed to perform live analysis of HTTP traffic: HACkSAw
- Tstat: fast and lightweight TCP analyser
  - Doesn't correlate requests and replies, misses the size of too many objects
- Bro: intrusion detection
  - Accurate, but memory hungry and too slow

Tool	Detected requests	CPU [s]*	Memory usage [GB]	0-length replies
Tstat2.4	2 531 210	445	0.3	1 128 109
bro	2 559 056	8033	4.2	424 355
HACkSAw	2 426 391	368	5.8	328 465

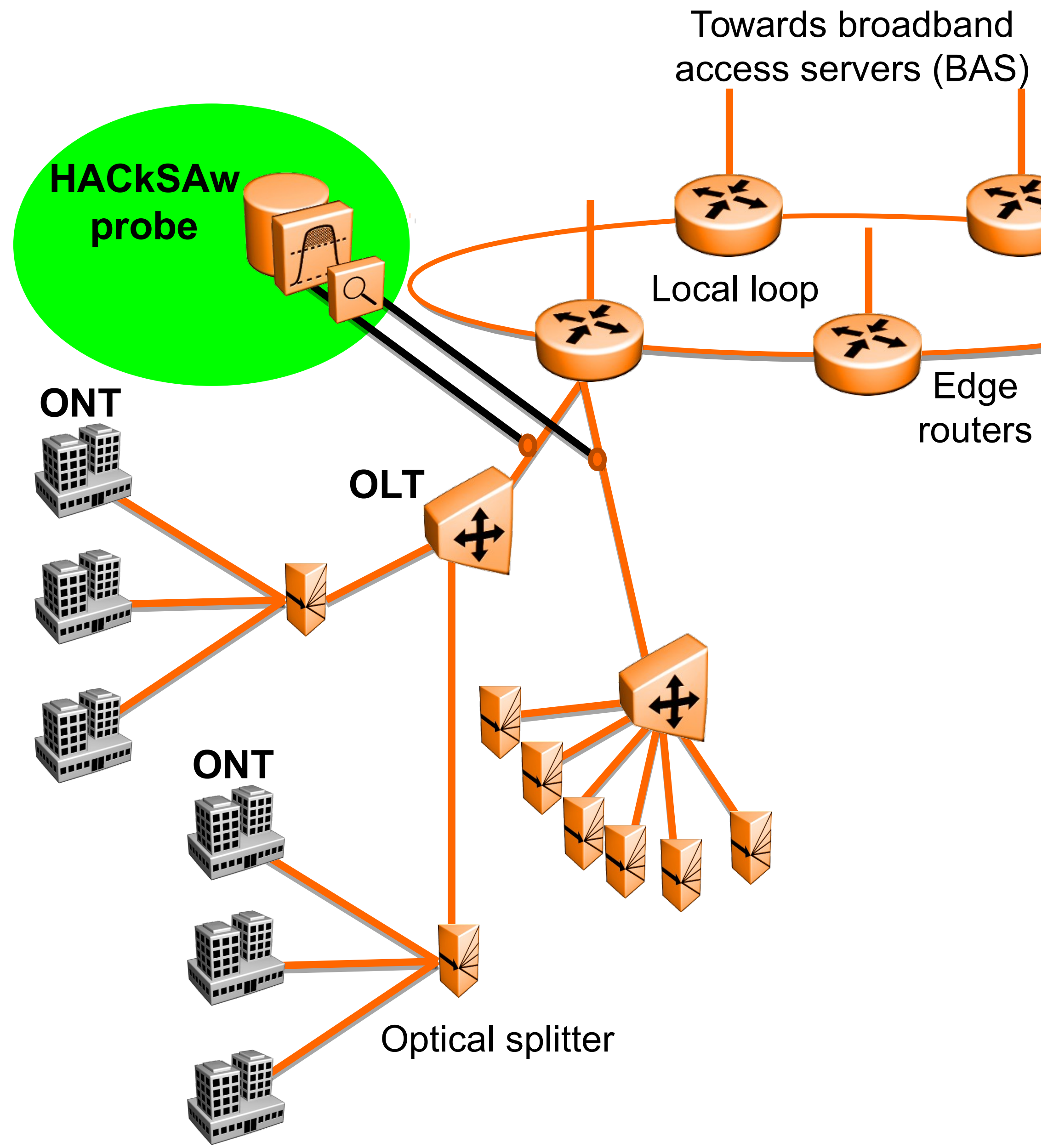
* Cumulative user time as reported by the *time* UNIX utility

Benchmark of our tool (HACkSAw) compared to 2 other popular tools
- 1-hour PCAP trace of mobile backhaul traffic, peak hour, several tens of thousands of users

Dataset and vantage point

- Residential users (mostly fiber)
- April 18th to May 30th 2014: 42 days (~1000 hours)
- Few users
  - Investigating potential advantages of caching with low user aggregation

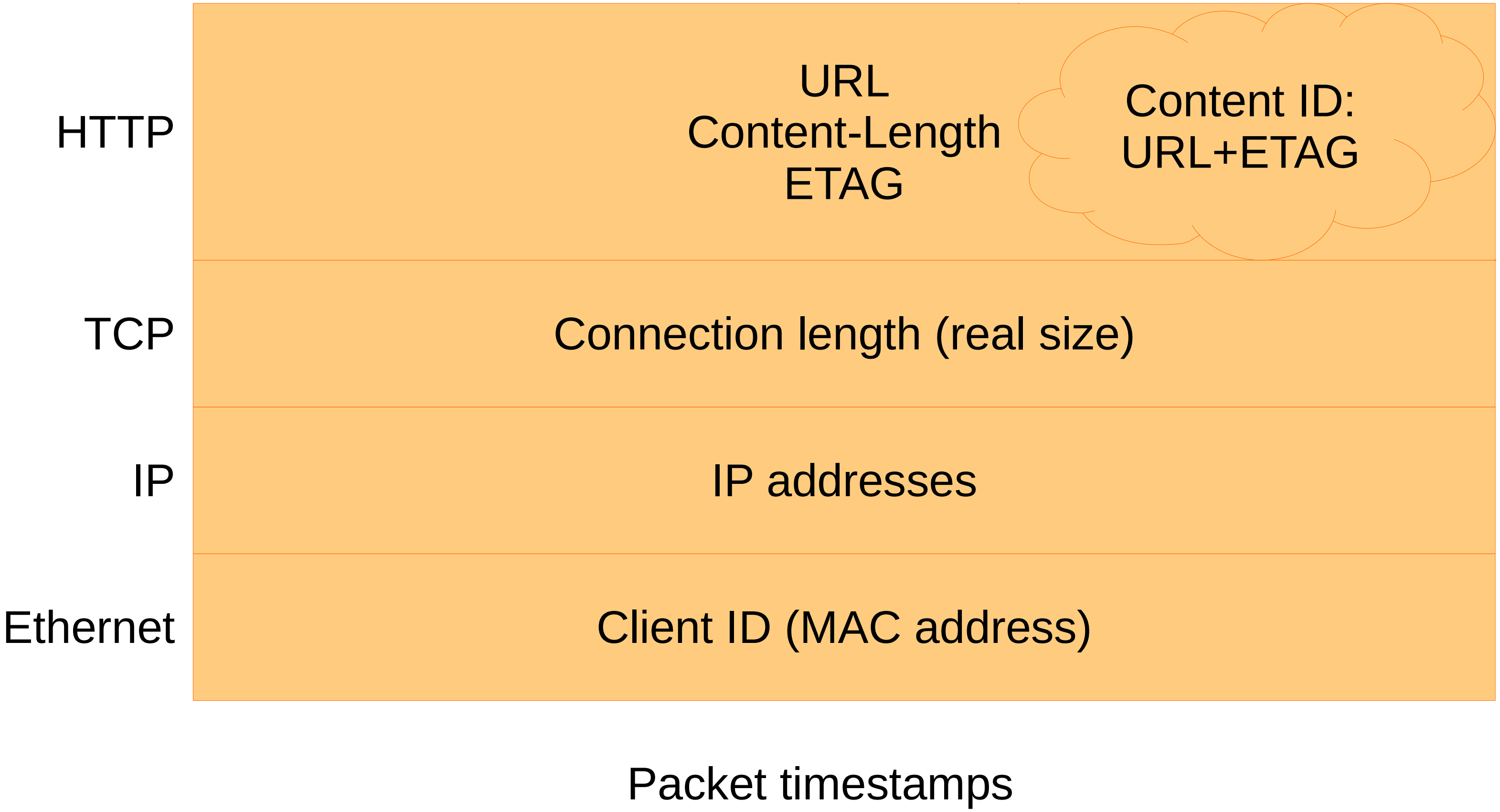
	Total	Daily average
Distinct users	1478	1218
HTTP requests	369 238 000	8 586 000
HTTP data volume	37 TB	881 GB
Distinct objects	174 283 000	4 956 000



Outline

- Motivation
- Monitoring
- **Collected statistics**
- Trace driven simulation
- Traffic characterization
- Conclusions

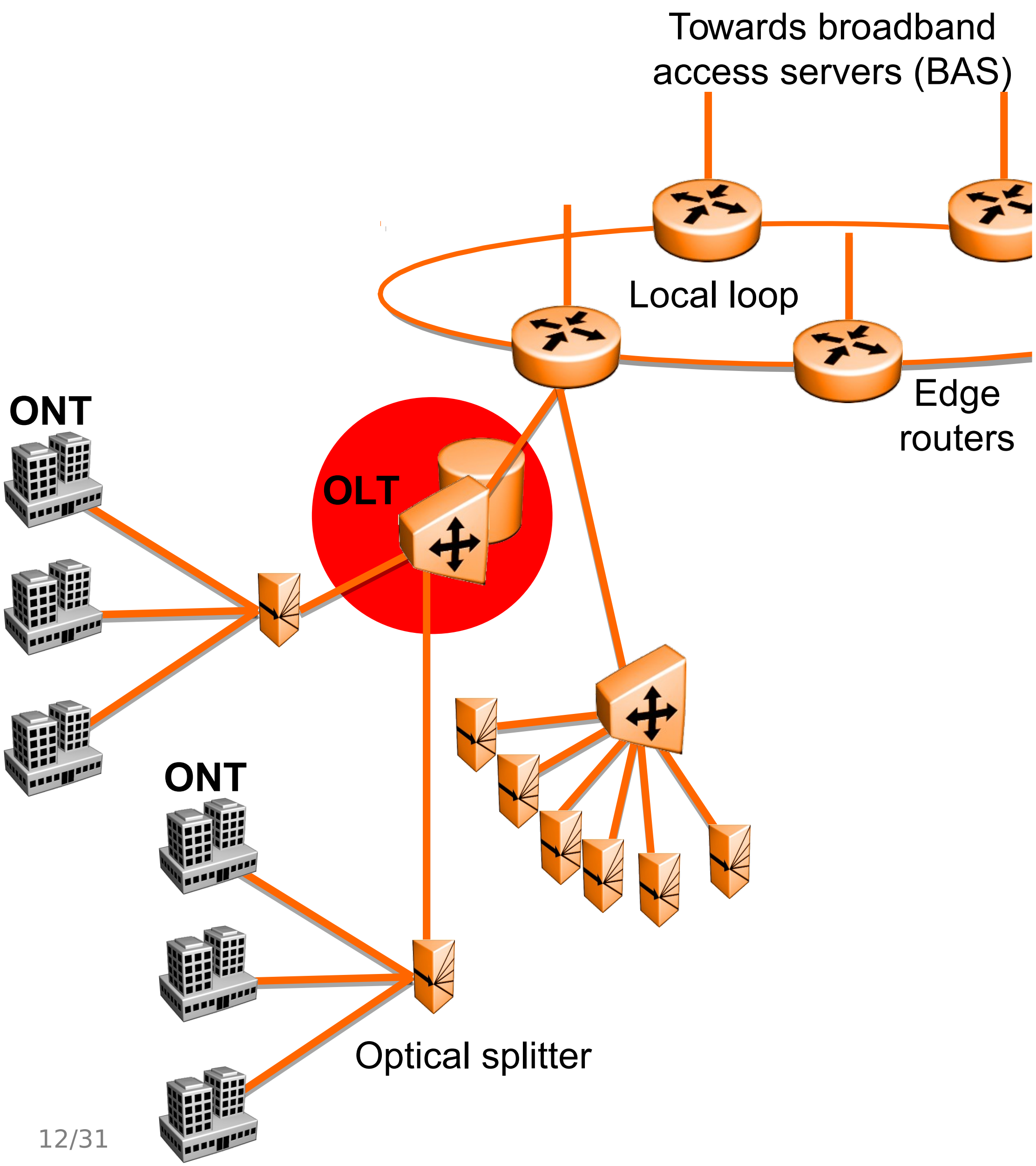
Collected statistics



Computed statistics

- Statistics refer to a specific time interval
  - Timestamp of request considered for the whole transaction
- Request cacheability R(t)
  - Percentage of repeated requests for the same object
- Traffic reduction T(t)
  - Percentage of traffic that could be cached by an oracle "perfect" cache with prefetching
- Virtual cache size V(t)
  - Sum of the size of all cacheable objects observed
  - Basically the minimum size needed for a "perfect" oracle cache

First scenario: cache at OLT



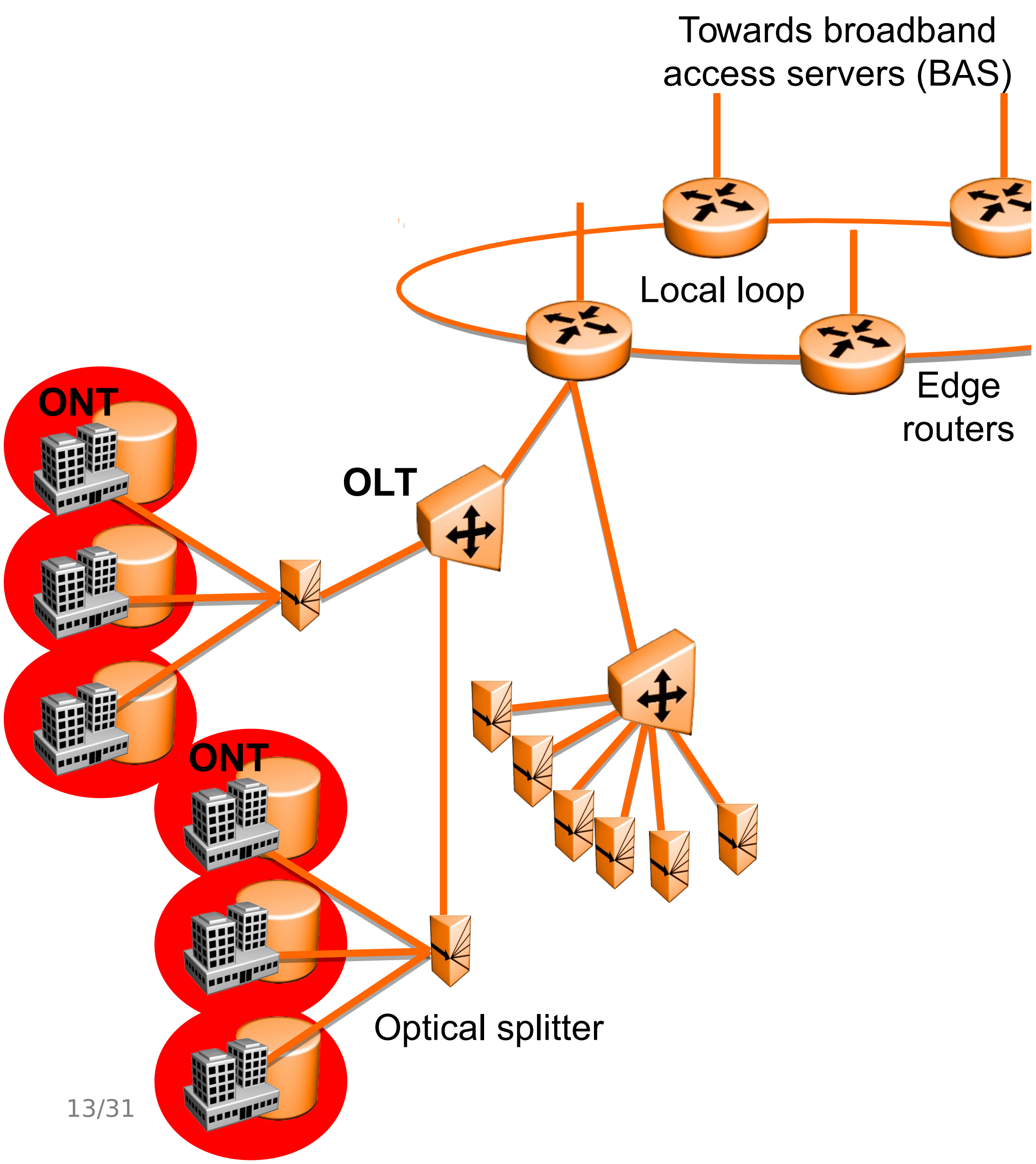
3 different scenarios:

1. **Cache at OLT only (vantage point)**

2 timescales:

- 1 hour
- 1 day

Second scenario: cache at ONT



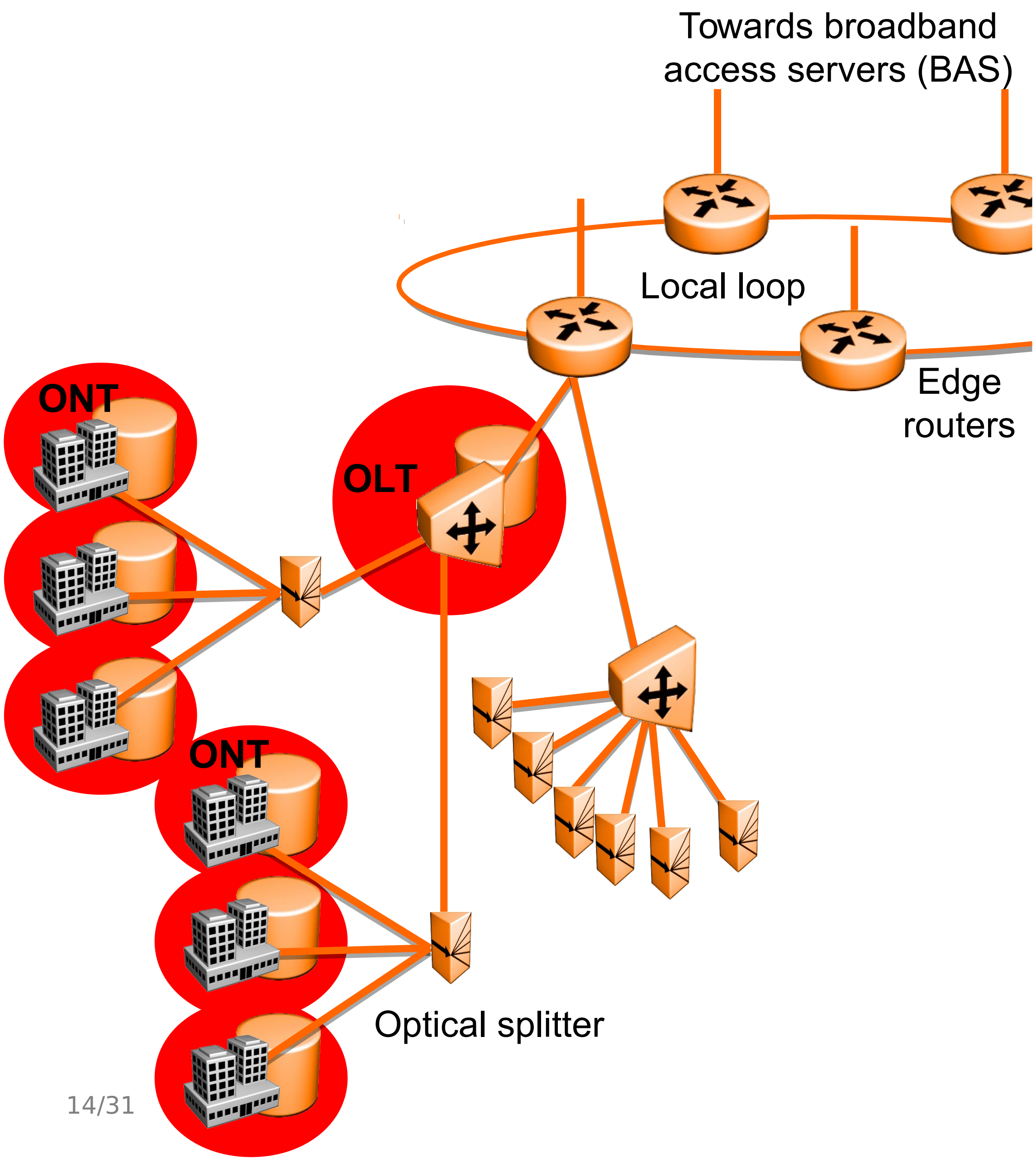
3 different scenarios:

2. **Cache at ONT only (users)**

2 timescales:

- 1 hour
- 1 day

Third scenario: cache at OLT and ONT



3 different scenarios:

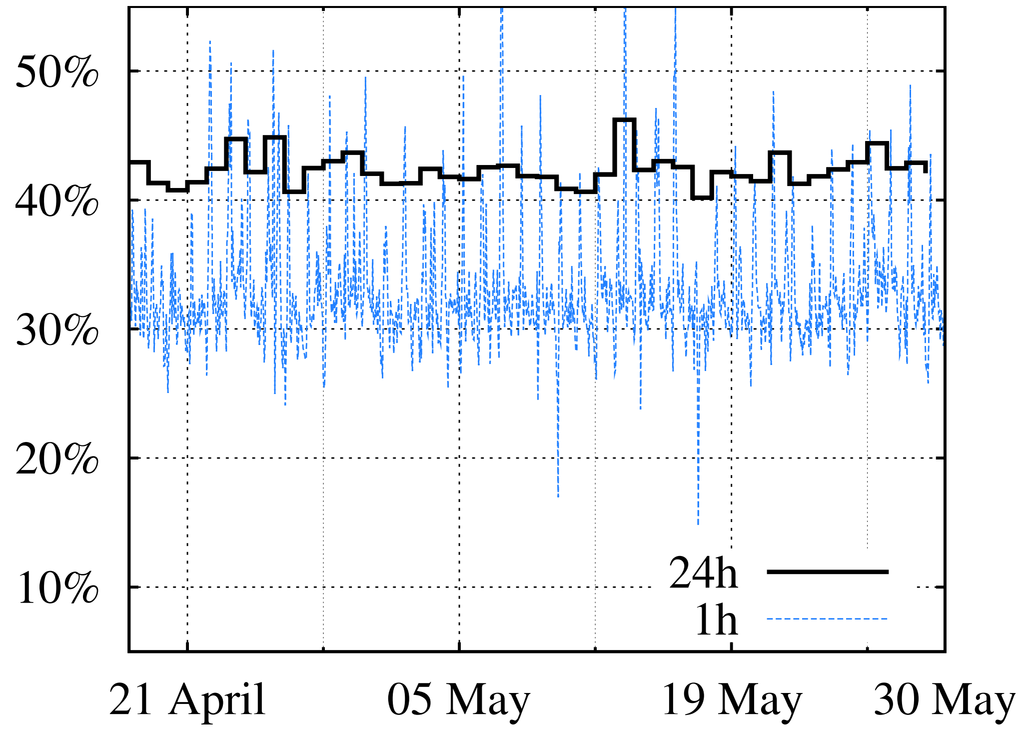
3. **Cache both at ONT and OLT**

2 timescales:

- 1 hour
- 1 day

Results: Cacheability and Traffic reduction – caching at OLT

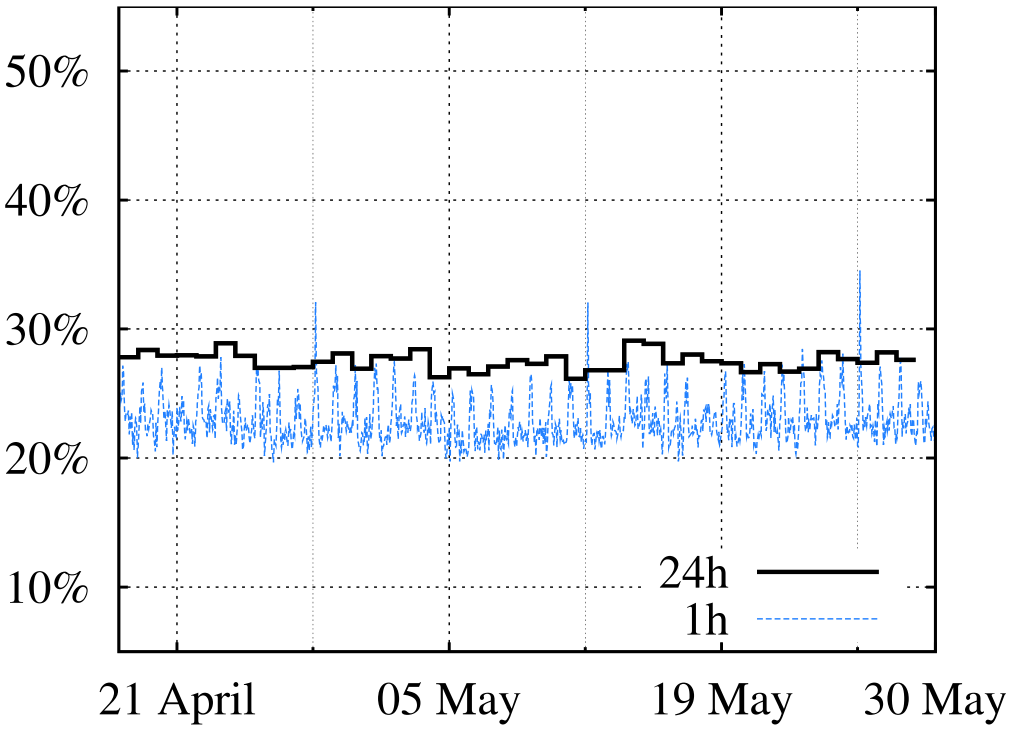
- Cacheability >40%; Traffic reduction around 35%
- Virtual cache size around 100GB
  - Feasible in a line card with current technology



Cacheability at OLT

Results: Cacheability and Traffic reduction – caching at ONT

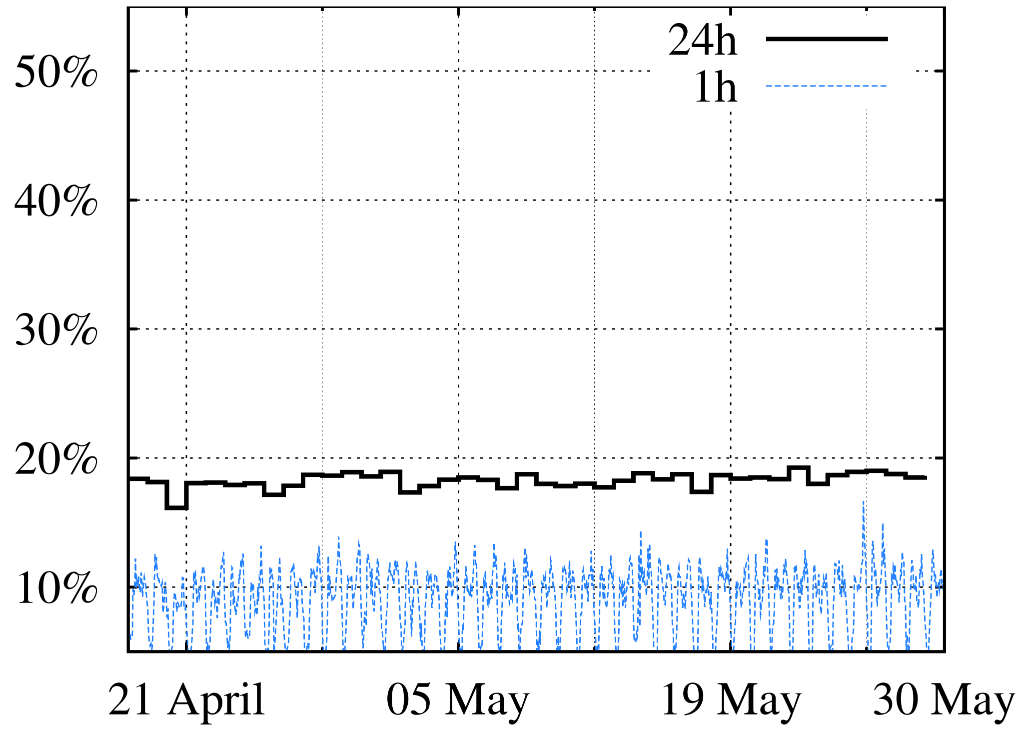
- Cacheability >25%; Traffic reduction >20%
- Virtual cache size around 100MB
  - Values are averages, high variance
  - Feasible in a home router with current technology



Cacheability at ONT

Results: Cacheability and Traffic reduction – caching at OLT and ONT

- Cacheability <20%; Traffic reduction around 20%
- Virtual cache size around 100GB
  - Feasible in a line card with current technology



Cacheability at OLT after ONT caches have filtered the traffic

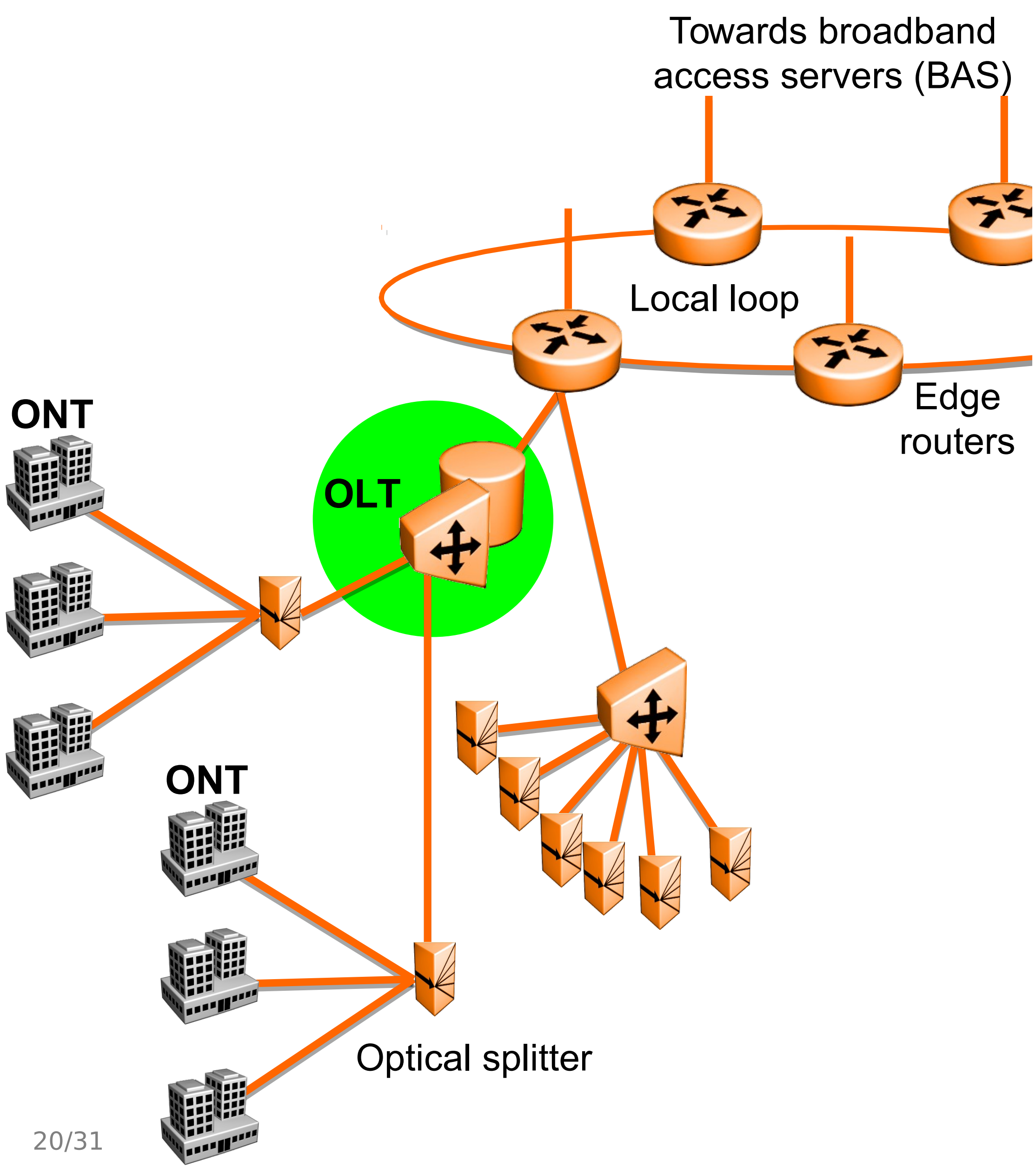
Summary of results

	OLT only	ONT only	OLT after ONT
Cacheability	42%	28%	19%
Traffic reduction	34%	24%	20%
Virtual cache size	100**GB**	100*MB*	100**GB**

Outline

- Motivation
- Monitoring
- Collected statistics
- **Trace driven simulation**
- Traffic characterization
- Conclusions

LRU simulation



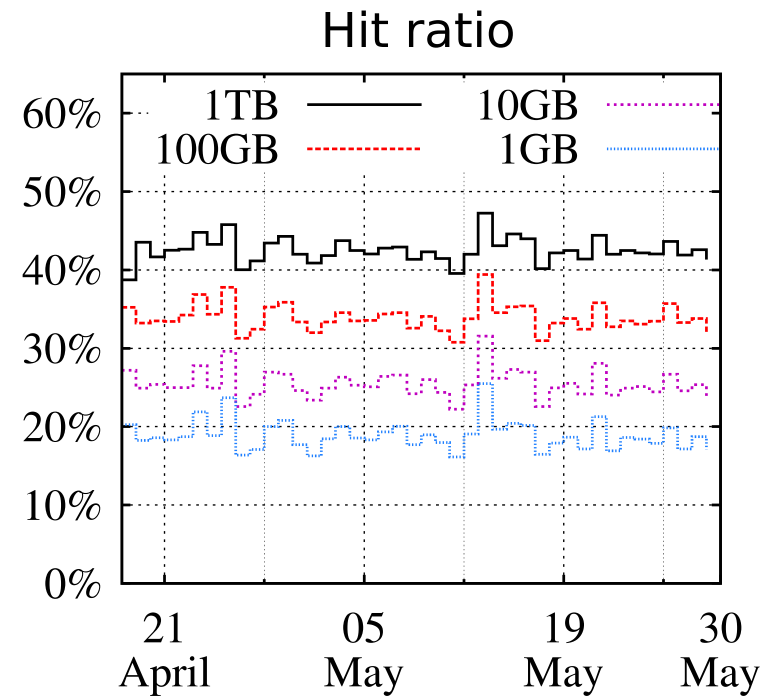
Trace driven simulation, only one scenario:

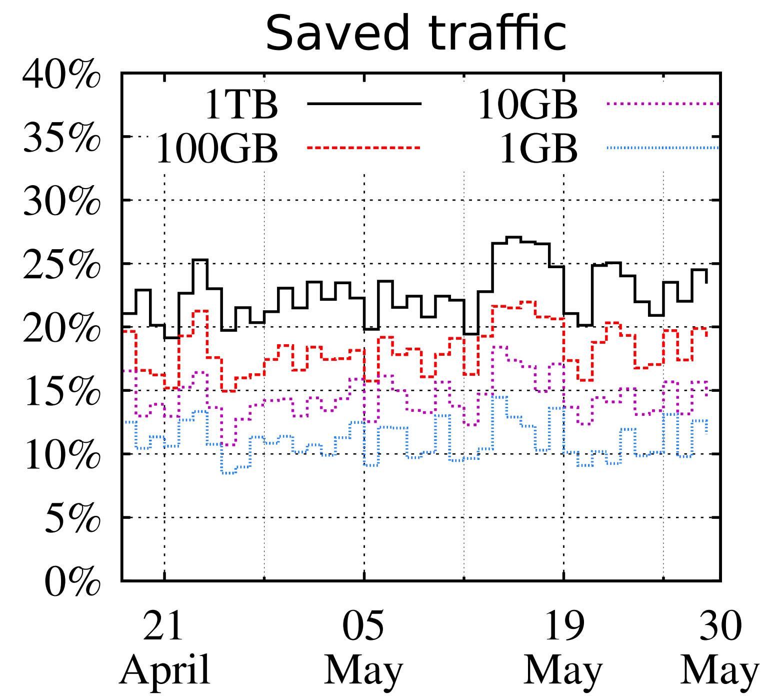
1. **Cache at OLT only (vantage point)**

- Needed to measure the performance of a real cache
- Needed to assess the impact of temporal locality
- Chunk-level cache: only the requested part of the objects is cached

Results: Simulated LRU

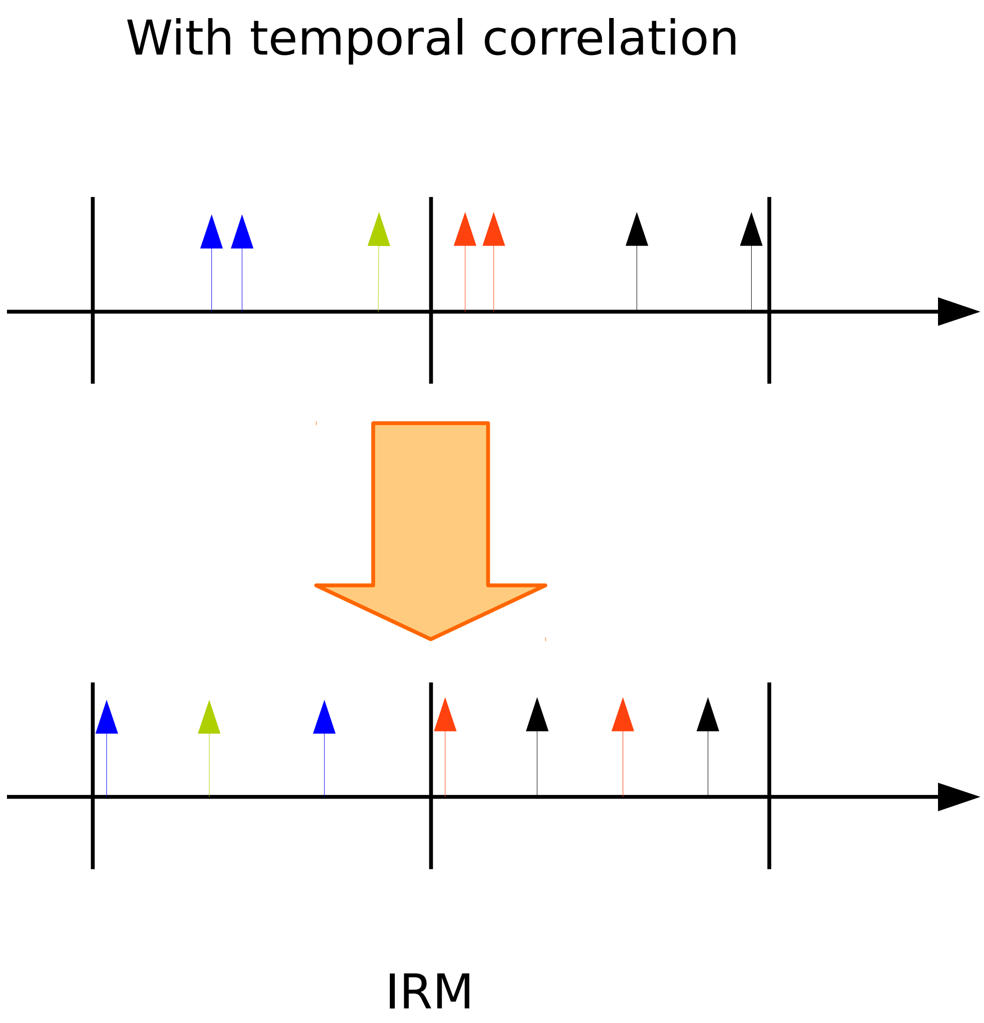
- 4 cache sizes: 1GB, 10GB, 100GB and 1TB
- Standard LRU replacement
- As expected the real performance is lower than the theoretic values shown before
  - Still acceptable
- Cacheability gives a rough estimation of the real values
  - Cacheability is easier to compute than doing an LRU simulation





LRU simulation – shuffling

- We want to assess the impact of temporal locality on the performance of the cache
- Shuffling: transactions are uniformly randomly redistributed inside their timeslots
- 1 hour and 1 day timeslots
- Smaller caches suffer, larger caches unaffected
  - Effects of temporal locality



LRU simulation – shuffling results

Hit ratio

Cache size	No shuffling	Shuffling: 1 hour slots	Shuffling: 1 day slots
1TB	42%	42%	42%
100GB	35%	35%	30%
10GB	25%	22%	17%
1GB	19%	15%	10%

Traffic reduction

Cache size	No shuffling	Shuffling: 1 hour slots	Shuffling: 1 day slots
1TB	23%	23%	23%
100GB	17%	17%	13%
10GB	14%	11%	5%
1GB	11%	5%	2%

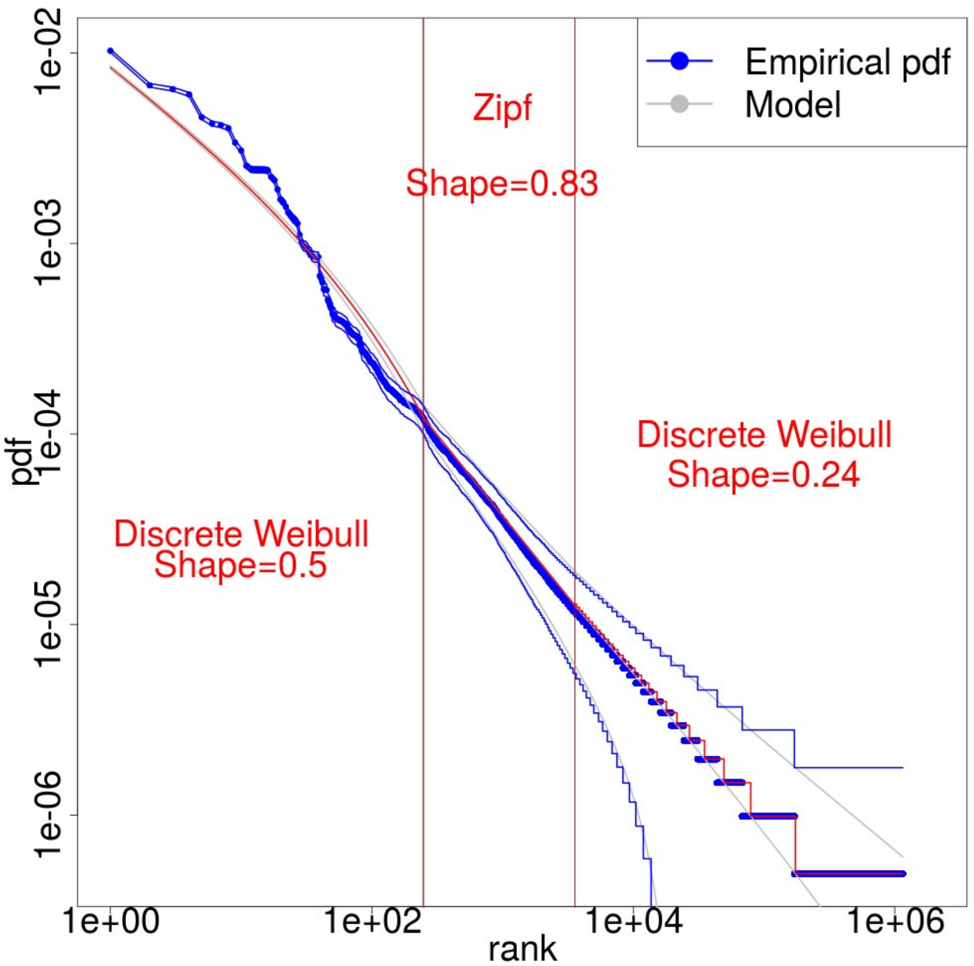
IRM is too conservative!

Outline

- Motivation
- Monitoring
- Collected statistics
- Trace driven simulation
- **Traffic characterization**
- Conclusions

Content popularity estimation

- Content popularity has a big impact on caching
- Zipf with α<1 has finite support
  - Confidence interval of active catalog size is unbounded
  - Catalog size is needed for correctly sizing the cache
  - Hit ratio $\propto \frac{\text{cache size}}{\text{active catalog}}$
- With a Weibull we can estimate all the parameters

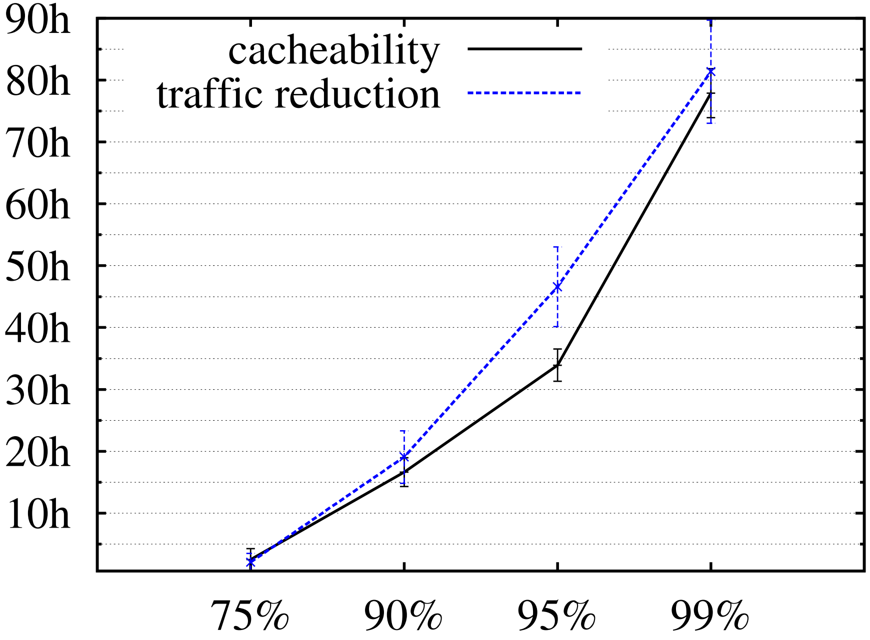


24 hours time period

95% confidence bands shown

Passes χ^2 goodness of fit

Time correlation



Average time needed to reach the percentiles of asymptotic cacheability and traffic reduction

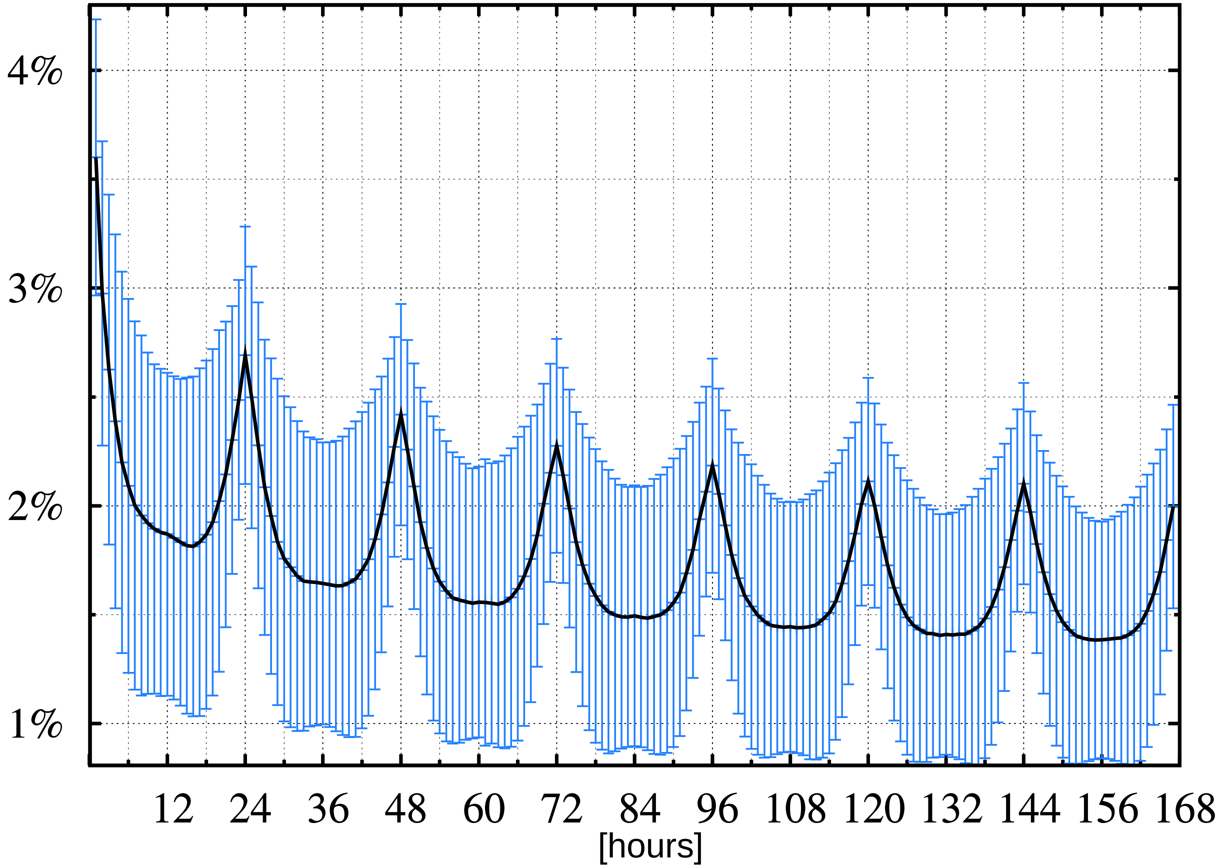
- What is the refresh rate of the active catalog?
  - Considering 1-hour timeslots, when and how much of the content in each timeslot will be requested again?
- Jaccard autocorrelation of the observed active catalog
  - Jaccard coefficient $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ hence $0 \leq J(A,B) \leq 1$
  - The autocorrelation function is hence

$$\mathcal{R}_{\Delta T}(k) = \frac{1}{n} \sum_{i,j:|i-j|=k}^{n} J(C_{i\Delta T}, C_{j\Delta T})$$

with $C_{i\Delta T}$ being the content catalog observed in the time window $i\Delta T$

Results: Jaccard autocorrelation

- Content changes quickly
- Weak 24-hours periodicity
- On average, less than 4% of the active catalog is the same in two consecutive 1-hour timeslots



$\mathcal{R}_{\Delta T}(k)$ for k in [1,168], ΔT=1hour, with standard deviation error bars

Outline

- Motivation
- Monitoring
- Collected statistics
- Trace driven simulation
- Traffic characterization
- **Conclusions**

Conclusions

- IRM Zipf models are inadequate, time correlation is more important
  - Problems estimating the size of the catalog
- Caching even in the access network yields real traffic savings
  - ICN solutions with native caching will provide real benefits
- Small caches (100MB on ONT and 100GB on OLT) can potentially provide a reduction of 35% of the traffic on the backhaul
- HTTPS = HTTP+TLS hinders all transparent caching efforts
  - So far only 15% of the traffic, but growing
  - ICN solutions ideal: security and caching are built-in primitives
  - New encryption-aware version of HTTP would be beneficial

Future work

- Further improvement of the tool
  - More accuracy, less memory consumption
  - Add support for more protocols
  - Low cost probes: tens of measurement points in Europe
- Repeat the measurements at different vantage points
  - More users, more traffic
- Include Layer 3-4 statistics
  - Latency
  - Throughput

Questions ?