Caching Algorithm for Content-Oriented Networks Using Prediction of Popularity of Content

Hiroki Nakayama[†], Shingo Ata[††], Ikuo Oka[††]
[†]BOSCO Technologies Inc.
[††]Osaka City University

Background

- Cache has an important roll to increase the performance of deployment service.
  - Reduces the load of server and increases the performance.
  - Contents Centric Network (CCN), Contents Delivery Network (CDN)
  - Algorithm such as Least Recently Used (LRU) and Least Frequently Used (LFU) causes waste usage of cache capacity.

- It is important to consider the popularity of each contents to use cache efficiently.

Related works

- Caching algorithm using content's popularity
  - WAVE: Popularity-based and Collaborative In-network Caching for Content-Oriented Networks (K. Cho, 2012)
  - Optimal Cache Allocation for Content-Centric Networking (Y. Wang, 2013)

- The fluctuations of popularity of content depending on time is not considered enough.
  - Only using the past popularity is difficult to handle the fluctuations of popularity.

⇨ Use predicted value of content's popularity

Objective

- Evaluate cache algorithm using prediction value of content's demand.
  - Decision of cache replacement is made by each cache nodes.
  - Content replacement is conducted based on prediction value.

- Predicting the popularity of every content is difficult.
  - The number of contents exceeds 2 billion[*1] and increasing day by day.
  - Propose the selection algorithm to decide which content should be predicted without lack of cache efficiency.

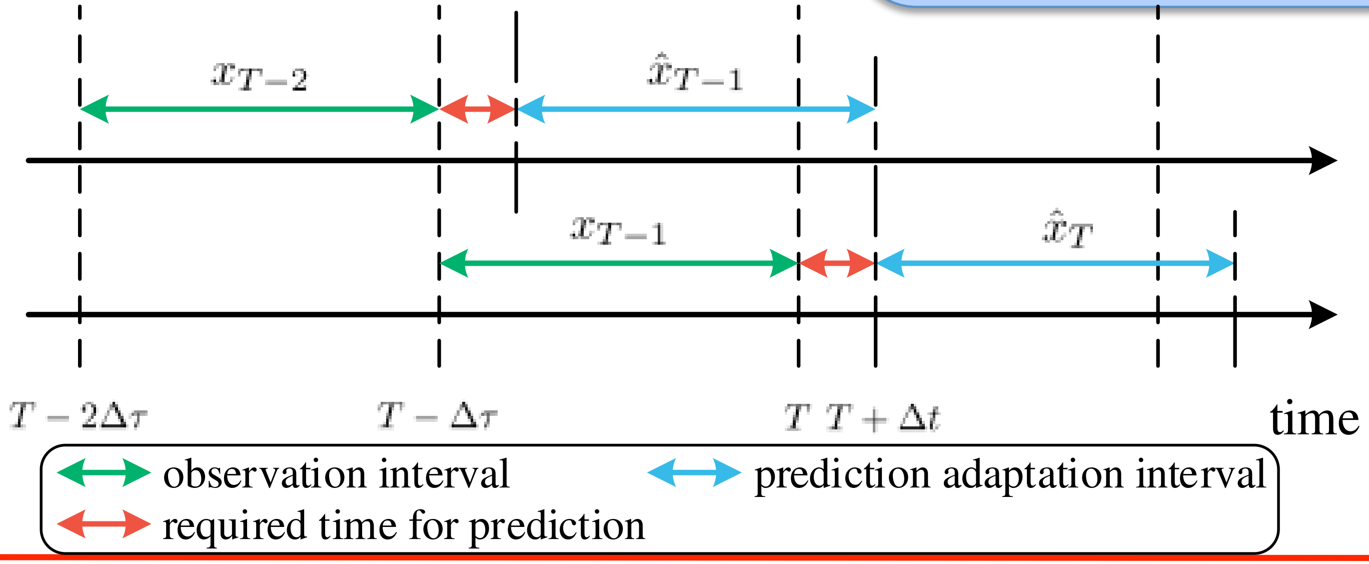
[*1] G. Gursun et al., "Describing and Forecasting Video Access Patterns," INFOCOM'12 March 2012

Caching algorithm using prediction

- We use the AutoRegressive (AR) model for prediction.
  - Burg method has been used to estimate AR coefficients.
  - p = 5 has been used for order of model.

$$\hat{x}_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \epsilon_t$$

x_t	:	trend data
α_i	:	AR coefficients
ϵ_t	:	residuals
p	:	order of parameters

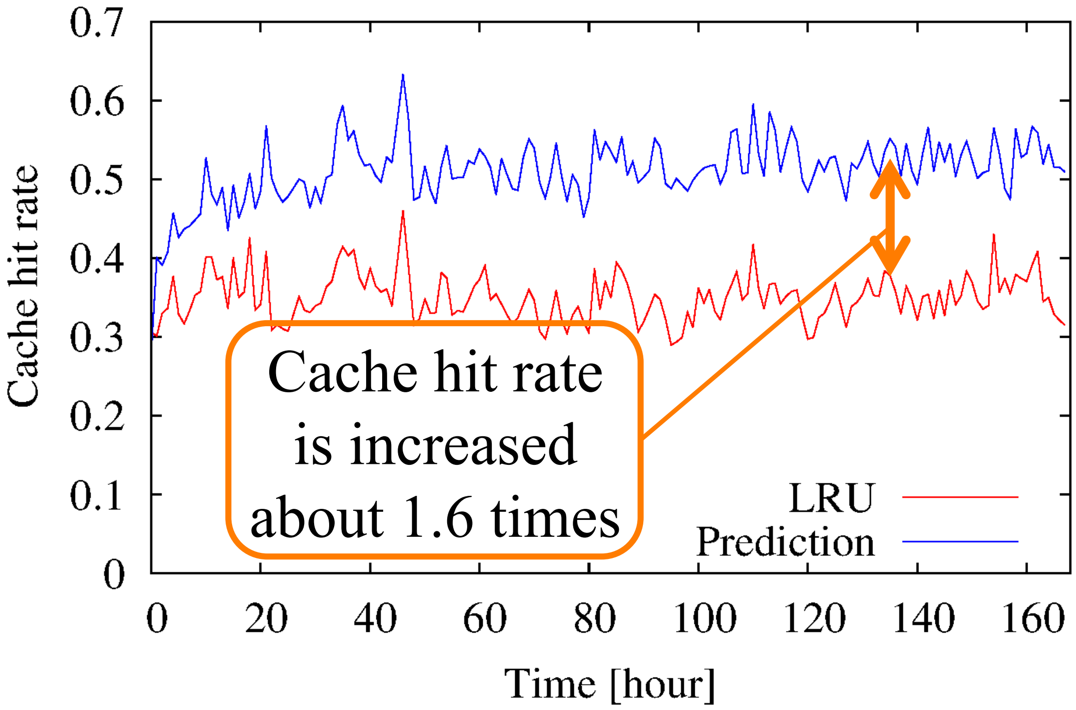


↔ observation interval ↔ prediction adaptation interval

↔ required time for prediction

Efficiency of using prediction

- Evaluate the cache hit rate in the CCN.



Cache hit rate is increased about 1.6 times

Parameter	Value	Unit
Requirement frequency	Avg 1	Hz
Amount of content	1,000,000	
Size of contents	Avg 10	MB
Size of chunk	0.1	MB
Cache capacity	10,000	MB
Number of cache node	46	
Prediction time unit (Δτ)	10	Min

- Using prediction is effective to increase cache usage.
  - Predicting whole content requires large computational costs.

Reduce the computational cost by limiting target to predict.

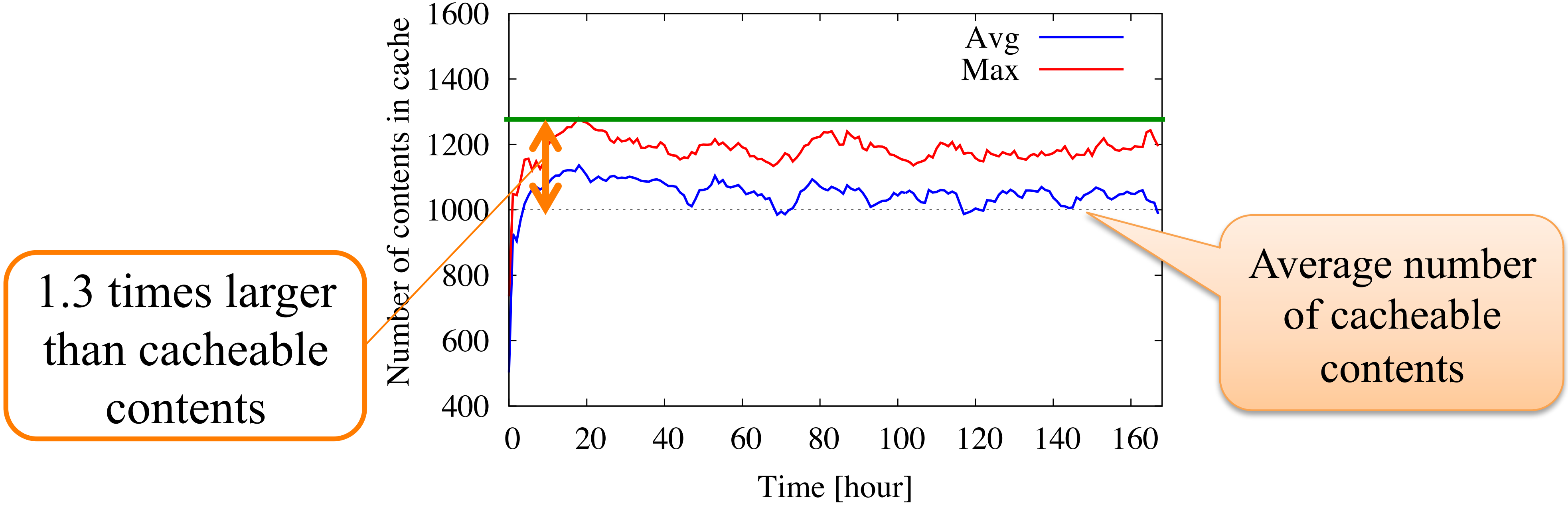
Reducing number of prediction target

- Most of the low demanded contents will never cached with the prediction cache.
  - Computational cost could be reduced simply by not predicting such contents.

- Key point on limiting number of prediction target.
  - Lack of performance by limiting prediction target should be minimized.
  - The number of contents which should be predicted.
  - The popularity of content will fluctuate rapidly.

Estimate number of prediction target

- Estimate the necessary number of contents to be predicted.
  - Corresponds to the number of contents cached at least once in the each time slot.



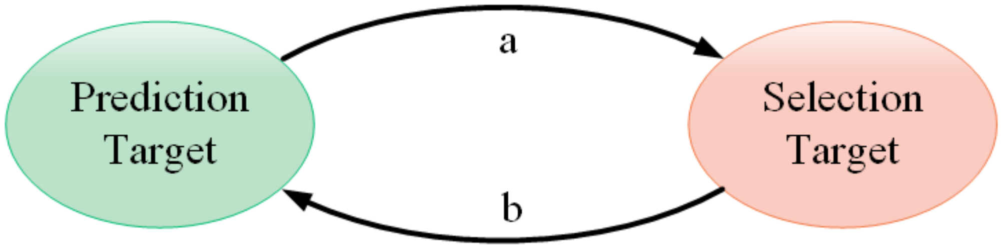
1.3 times larger than cacheable contents

Average number of cacheable contents

- The popularity of contents could change rapidly.
  - Simply limit the prediction target to twice of average number of cacheable contents.

Target limitation with 2 observation states

- Use 2 observation states for prediction target limitation.



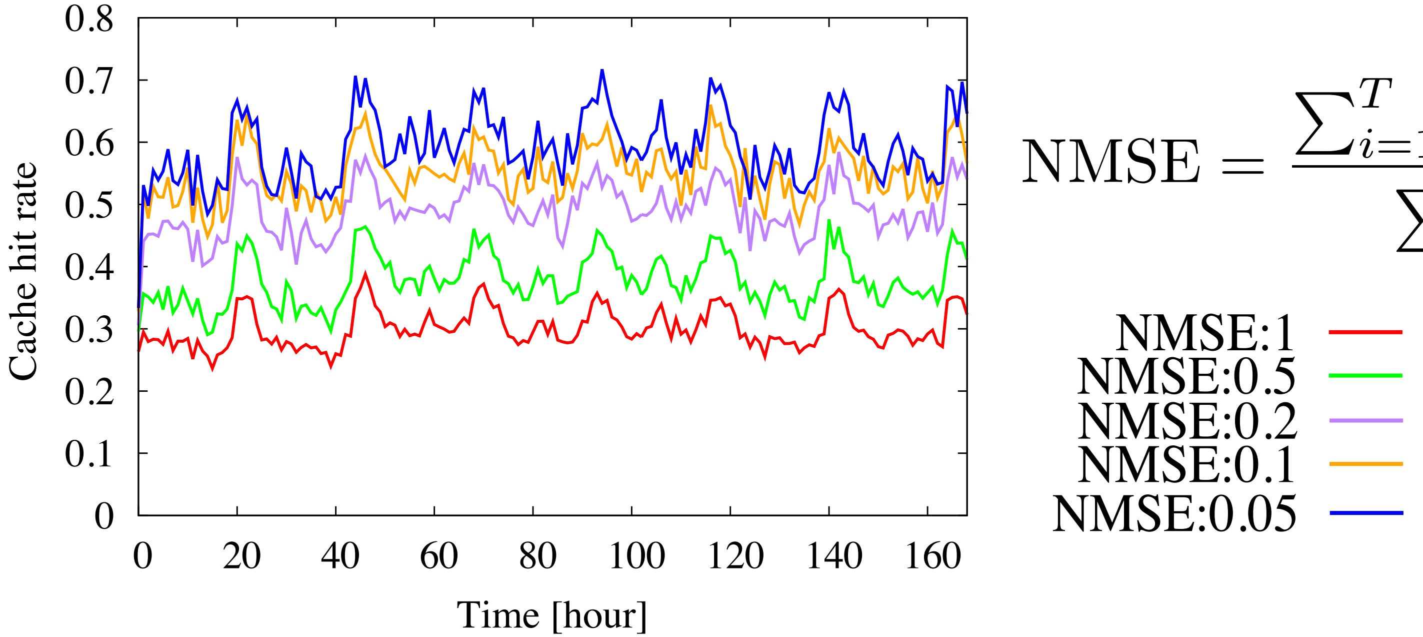
- Prediction Target (PT)
  - Contents which will be predicted and cached.
- Selection Target (ST)
  - Unpredicted contents. LFU is used for handling.

	Transit condition
a	The content that is not cached for specific time will be transit.
b	Transit frequently required contents based on LFU.

By only predict the contents belonging to PT,
the computational cost will be reduced greatly.

The influence of prediction error

- The contents transited from ST does not have enough popularity.
  - It causes an inaccurate estimation of prediction model and large prediction error.
- Evaluate how the prediction error affects.

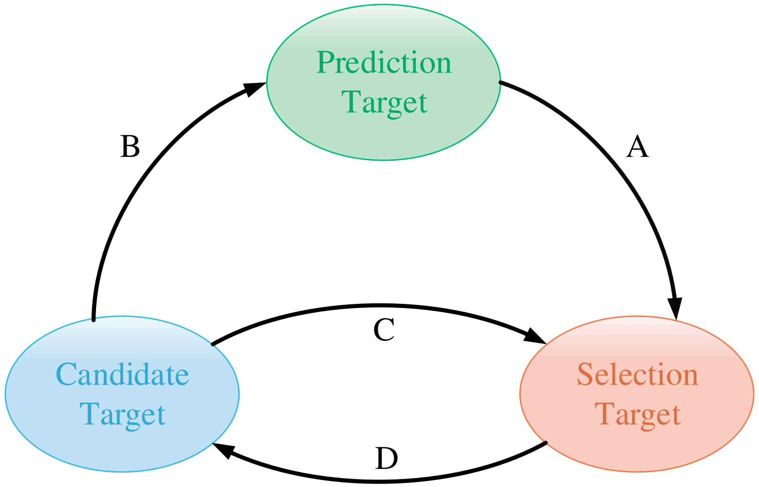


$$\mathrm{NMSE} = \frac{\sum_{i=1}^{T}(x_i - \hat{x_i})^2}{\sum_{i=1}^{T} x_i^2}$$

- Prediction error makes less cache performance.

Use state that observe the popularity of content with time series

Target limitation with 3 observation states



- Prediction Target
  - Contents which will be predicted and cached.
- Candidate Target
  - Contents to observe the popularity as time series.
- Selection Target
  - Unpredicted contents. LFU is used for handling.

	Transit condition
A	The content that is not cached for specific time will be transit.
B	The most popular content will be transit.
C	The content that has not been transit to PT will be transit.
D	Transit frequently required contents based on LFU.

Simulation environment

- Prepared CCN simulator based on ccnSim[*1].
  - Implemented caching algorithm including ProbCache[*2] and WAVE.
- Prepare YouTube-like simulation environment[*1].

Parameter	Value	Unit
Requirement frequency	Avg 1	Hz
Amount of content	1,000,000	
Size of contents	Avg 10	MB
Size of chunk	0.1	MB
Cache capacity	10,000	MB
Number of cache node	46	
Prediction time unit (Δτ)	10	Min

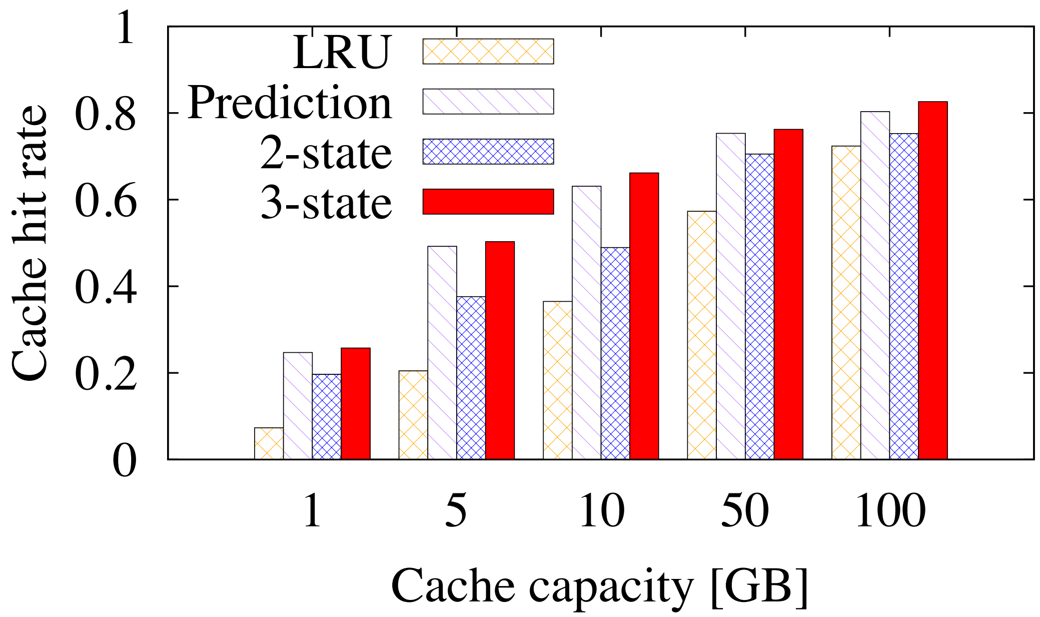
- The real monitored data is used for content's popularity.
  - Collected query messages in Gnutella as the data set.
  - The measurement was conducted from 2012/4/26 to 2012/9/12.

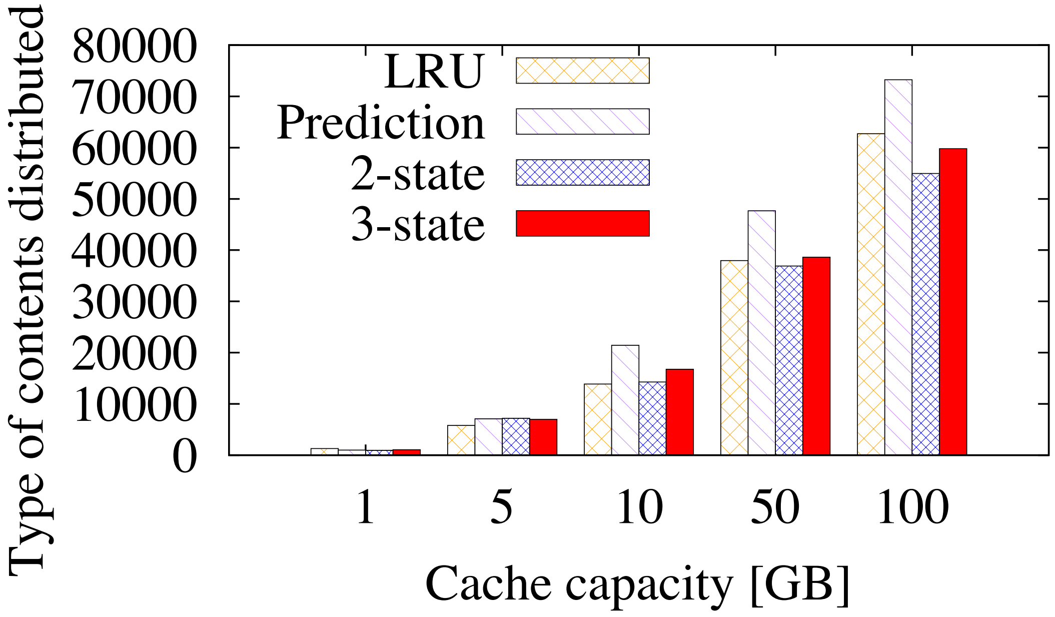
[*1] D. Rossi et al., "Caching performance of content centric net- works under multi-path routing (and more),"

[*2] I. Psaras et al., "Probabilistic In-Network Caching for Information-Centric Networks," ICN'12, August 2012

[*3] N. Spring et al., "Measuring ISP topologies with Rocketfuel," SIGCOMM 2002, August 2002

Performance evaluation for selection algorithm

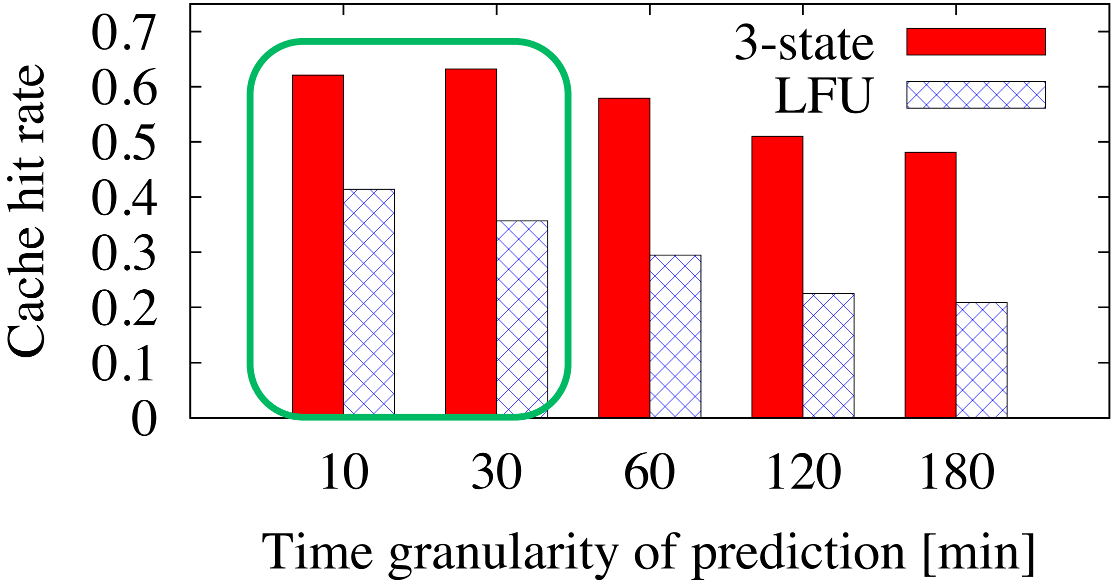




- 2 state selection algorithm occurs a lack of cache hit rate where 3 state selection algorithm does not.
  - Increasing prediction accuracy by using observation period is important.
- There are lack on the number of distributed content type by using selection algorithm.
  - Selection algorithm prevents to cache unpopular contents.

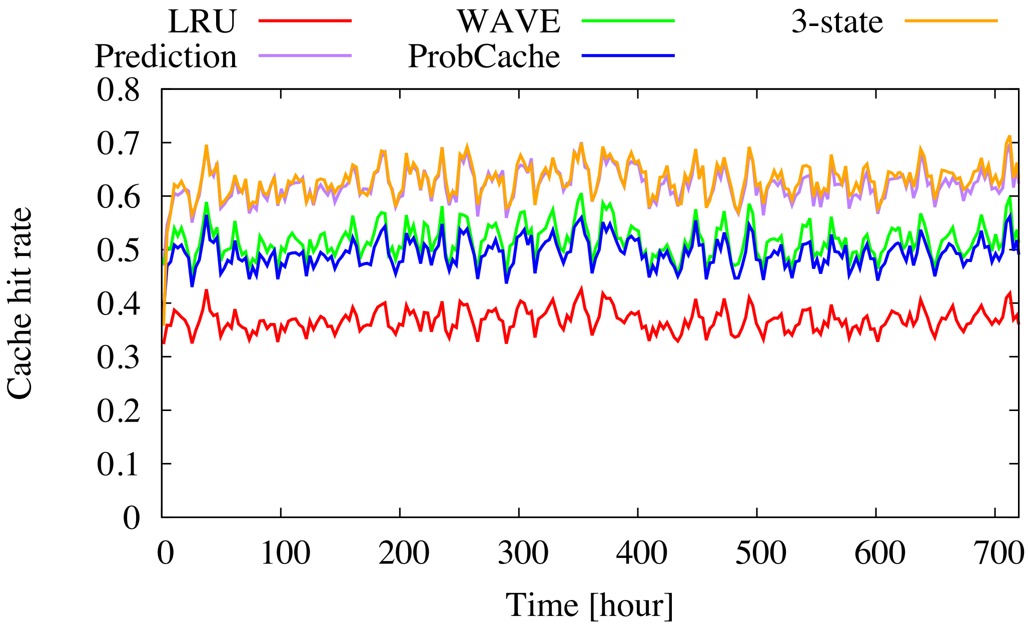
The influence of prediction time window size

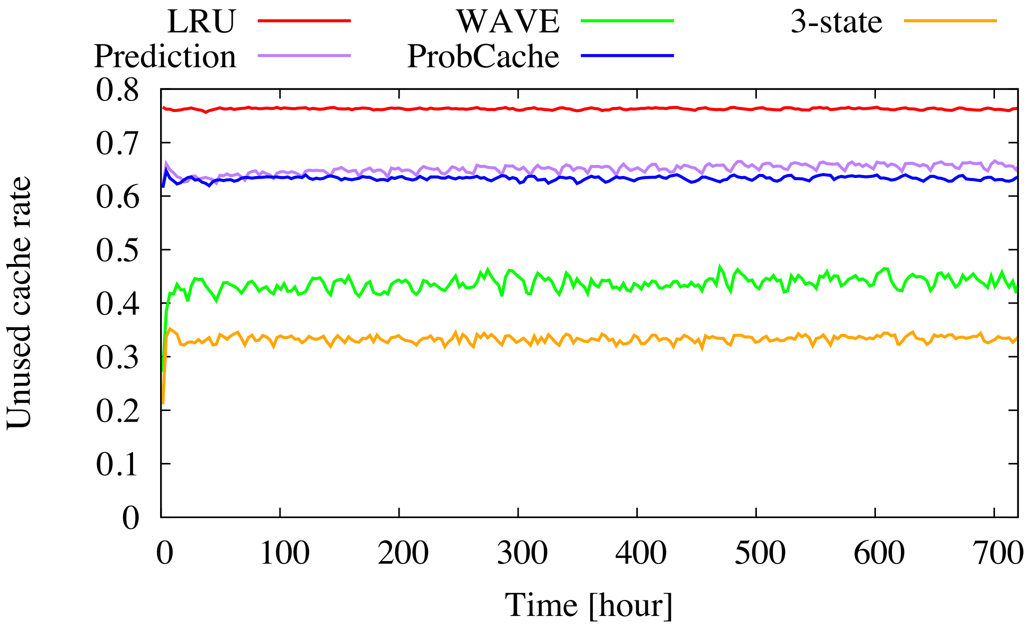
- The performance of selection algorithm could change due to the interval of updating prediction.



- There is no big difference between the cache hit rate of 10 minutes and that of 30 minutes.
- Not only considering the popularity but also predicting is important to gain performance.

Comparison evaluation with existing algorithm





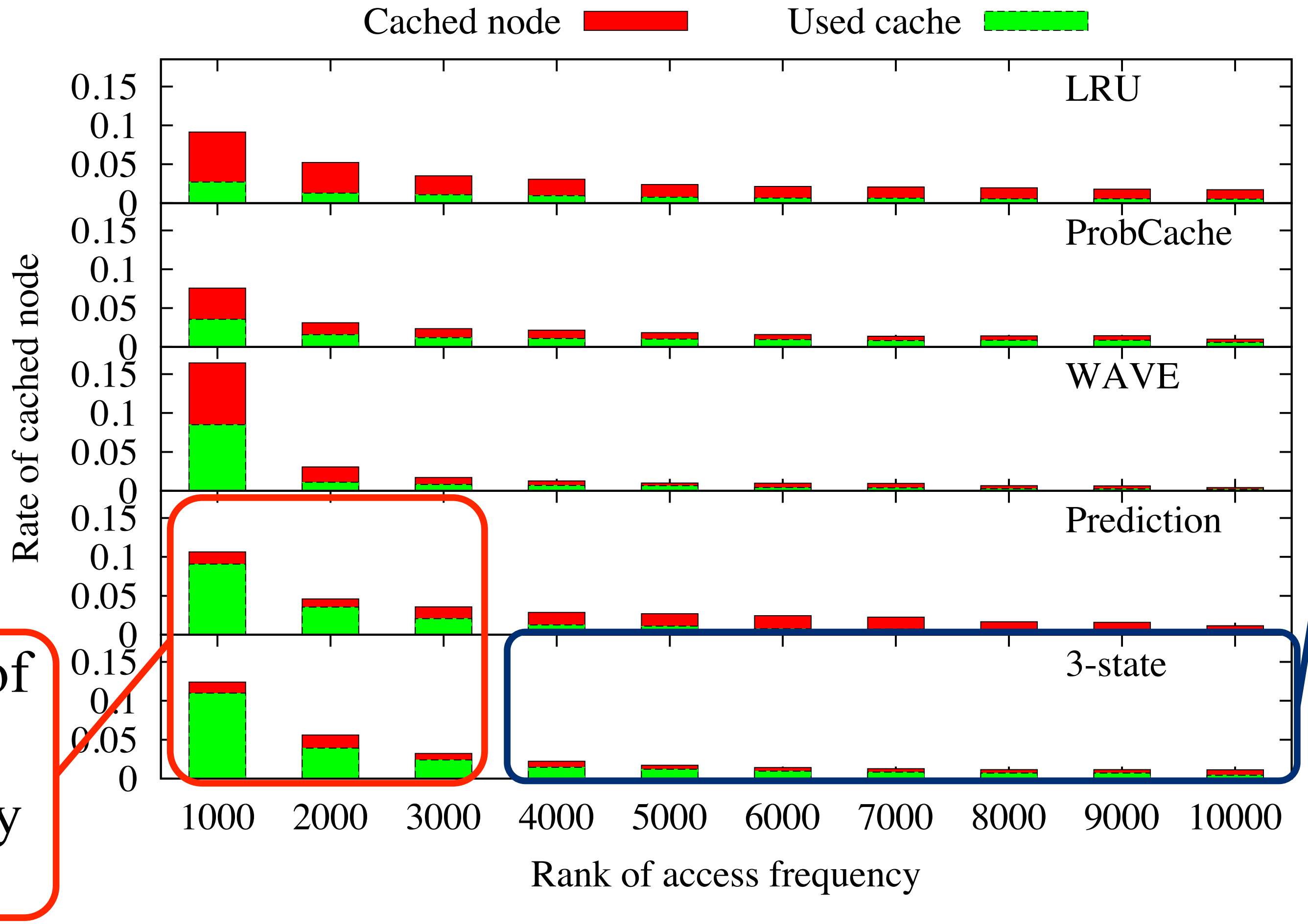
- Cache hit rate is increased by about 1.2 times comparing with existing methods.

- The unused cache will be reduced by using selection algorithm.
  - Selection algorithm can avoid to cache unpopular contents.

The usage of cache became more efficient by using prediction and selection algorithm.

Evaluation focused on content's popularity

- Popular contents should be cached on several cache nodes.
  - It is important on the aspect of efficient usage of cache.



The usage of cache increases by prediction

Influence of prediction error are reduced by selection algorithm

Conclusions & future works

- Conclusions
  - We have introduced the cache algorithm using prediction and selection algorithm to increase performance of cache with small computational cost.
  - We have demonstrated that the cache hit rate is increased about 1.6 times.
  - The result also indicates that our proposal method is effective on reducing number of unused cache.
- Future works
  - Evaluate the performance by using other prediction model.
  - More evaluation on computational cost.

Caching algorithm using prediction

- We use the AutoRegressive (AR) model for prediction.
  - Burg method has been used to estimate AR coefficients.
  - p = 5 has been used for order of model.

$$x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \epsilon_t$$

x_t	:	trend data
α_i	:	AR coefficients
ϵ_t	:	residuals
p	:	order of parameters

- The chunk of content will be cached based on its popularity.
  - Chunk is the part of content.
  - Demand of each chunk is based on that of content.
  - Cache with smallest value will be replaced.

Simulation environment

- Prepared CCN simulator based on ccnSim[*1].
  - Implemented caching algorithm including ProbCache[*2] and WAVE.
- Prepare YouTube-like simulation environment[*1].

Parameter	Value	Unit
Requirement frequency	Avg 1	Hz
Amount of content	1,000,000	
Size of contents	Avg 10	MB
Size of chunk	0.1	MB
Cache capacity	10,000	MB
Number of cache node	46	
Prediction time unit (Δτ)	10	Min

- The real network topology that is publicly available through Rocketfuel[*3] is used.

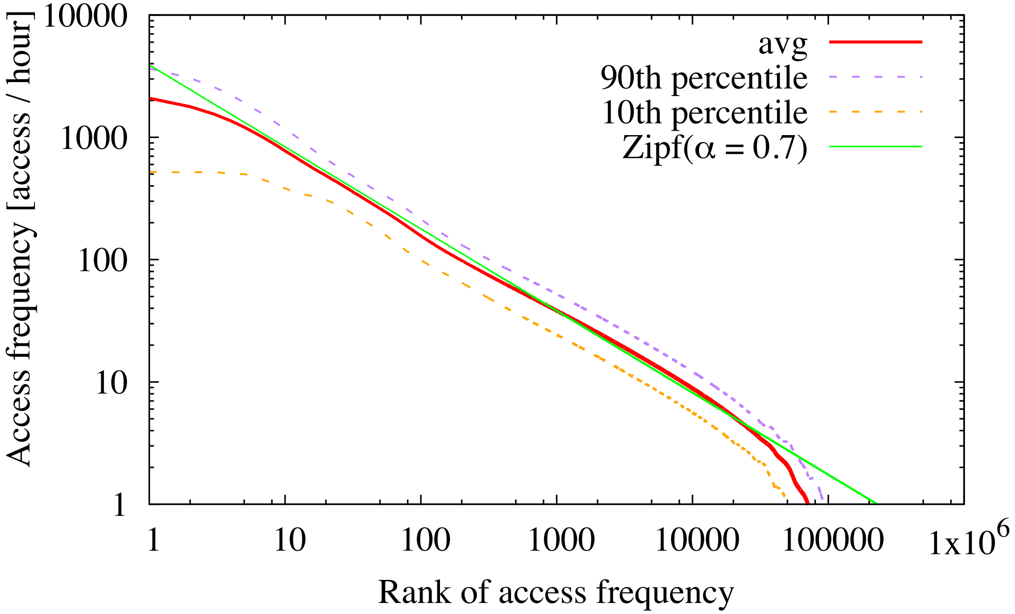
[*1] D. Rossi et al., "Caching performance of content centric net- works under multi-path routing (and more),"

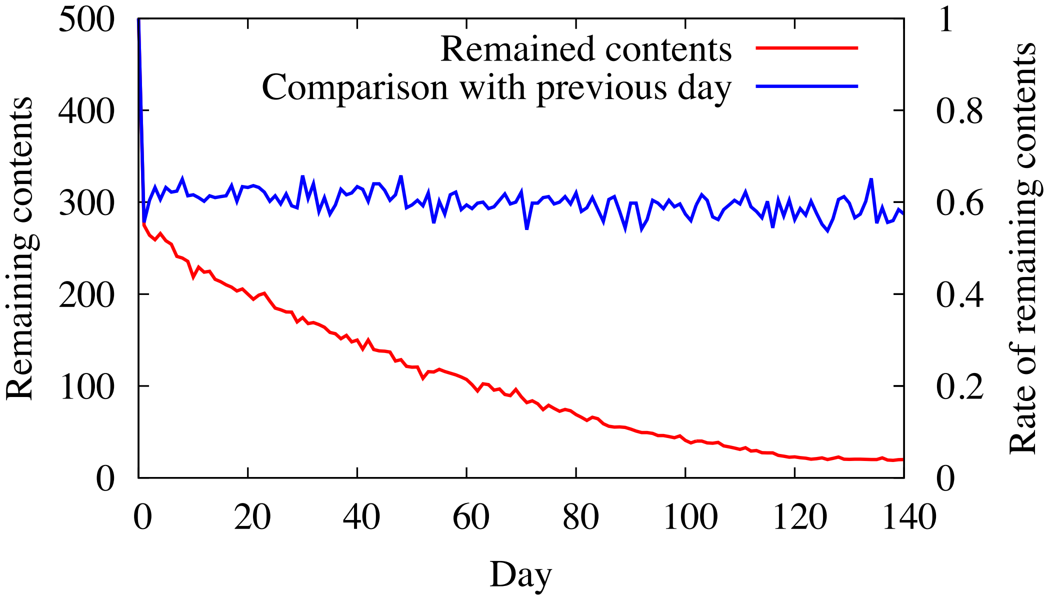
[*2] I. Psaras et al., "Probabilistic In-Network Caching for Information-Centric Networks," ICN’12, August 2012

[*3] N. Spring et al., "Measuring ISP topologies with Rocketfuel," SIGCOMM 2002, August 2002

Dataset

- The real monitored data is used for content's popularity.
  - Collected query messages in Gnutella as the data set.
  - The measurement was conducted from 2012/4/26 to 2012/9/12.





$$f(k) = \frac{\frac{1}{k^{\alpha}}}{\sum_{n=1}^{N}\left(\frac{1}{n^{\alpha}}\right)}$$

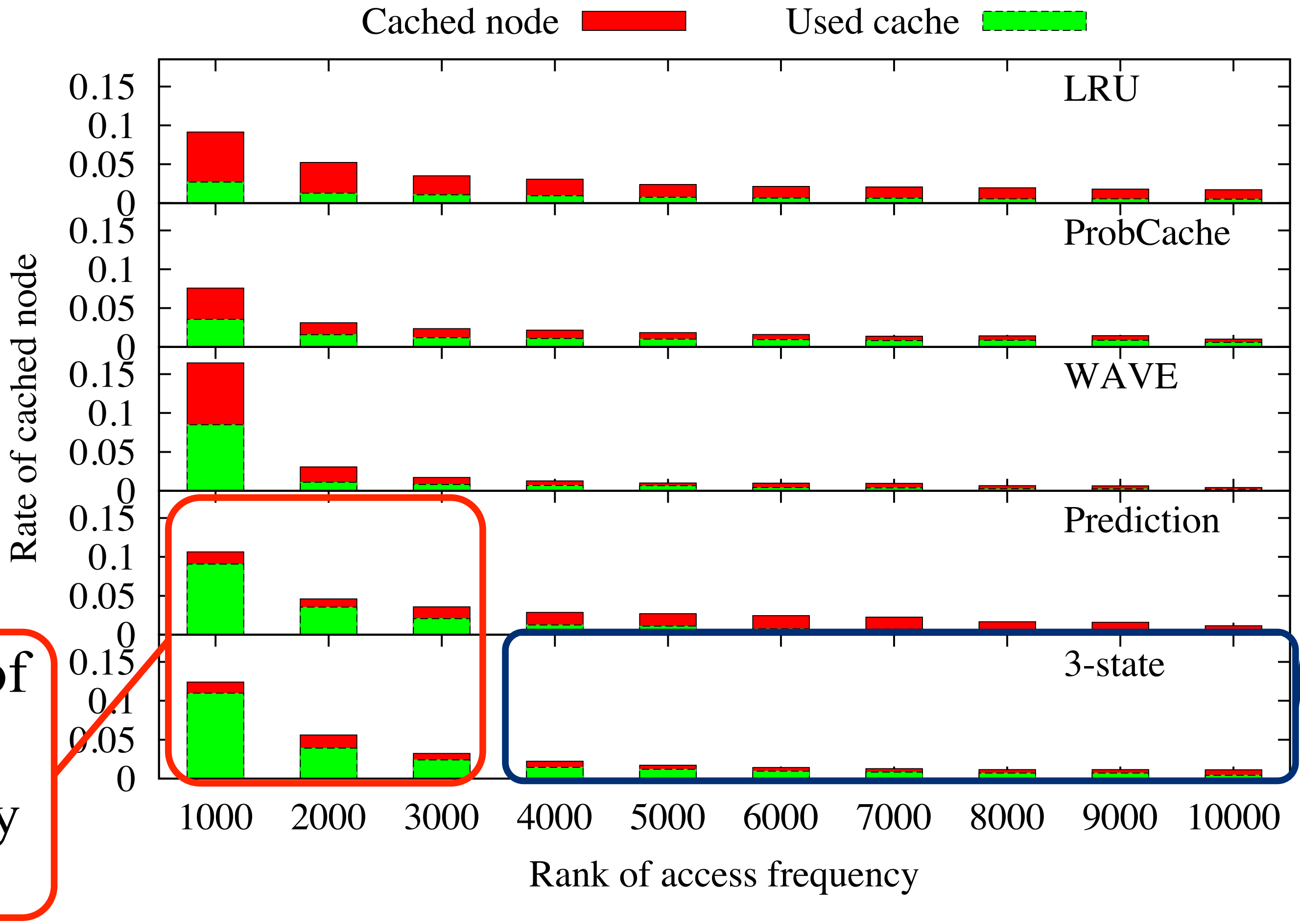
k : Rank of content

N: Number of content

The popularity changes rapidly

Evaluation focused on content's popularity

- Popular contents should be cached on several cache nodes.
  - It is important on the aspect of efficient usage of cache.



The usage of cache increases by prediction

Influence of prediction error are reduced by selection algorithm