## Adaptive Media Streaming on P2P Networks

Masahiro Sasabe, Naoki Wakamiya, Masayuki Murata
Osaka University
m-sasabe@cmc.osaka-u.ac.jp

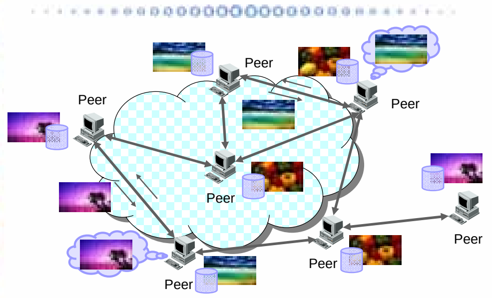
## Content





## Backgrounds

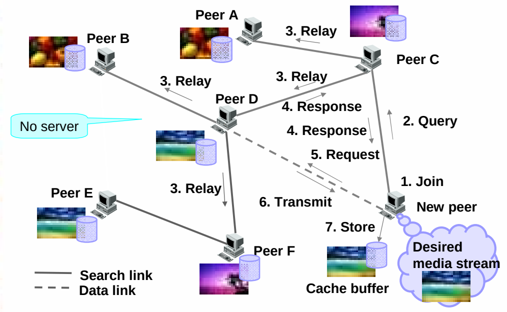
### - P2P (Peer-to-Peer) architecture -




P2P is a network paradigm to solve problems in the client-server architecture



## Proposed Media Streaming on Pure P2P Networks






## Previous Work

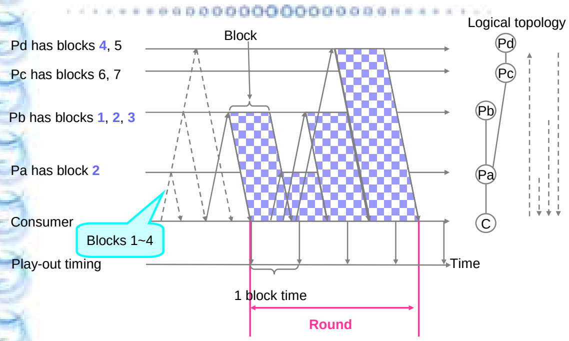
- We proposed efficient mechanisms for on-demand media streaming on pure P2P networks [7]
  - Segmentation of media stream
    - For efficient use of network bandwidth and cache buffer
  - Adaptive block-search method
    - To solve the scalability problem of search on pure P2P networks
  - Provider peer determination algorithm
    - To achieve continuous media play-out

[7] M.Sasabe, N. Wakamiya, M, Murata, and H. Miyahara, "Scalable and Continuous Media Streaming on Peer-to-Peer Networks," in *Proceedings of P2P2003*, pp. 92-99, Sept. 2003



## Basic Mechanism of Proposed Media Streaming

### - per-group block search and retrieval -

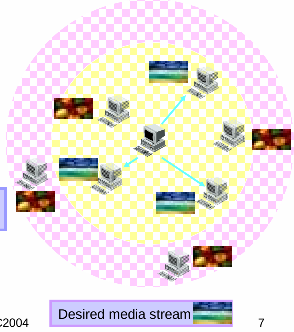


## Adaptive Block Search

- Full Flooding
  - Flooding with static value of TTL
- Limited Flooding
  - Flooding with limited TTL based on the search results at the previous round
- Selective Search
  - Send queries to particular peers based on the search results at the previous round

Consider pros and cons of them

- FLS method
  - Combination of full flooding, limited flooding, and selective search

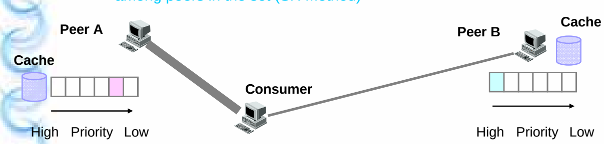





## Provider Peer Determination Algorithm

- Every time a peer receives a response message, it appropriately determines a provider peer for every block in the current round.

1. Calculate a set of peers from which it can retrieve a block in time.
2. Select a peer from the set
   a. Select a peer whose estimated retrieval time is the smallest among peers in the set (SF Method)
   b. Select a peer with the lowest possibility of block disappearance among peers in the set (SR Method)

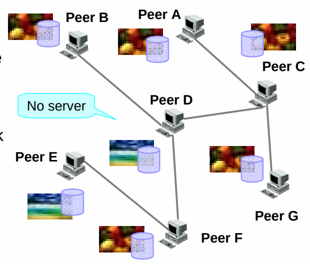





## Problem Found in Our Previous Work

- If LRU (Least Recently Used) is used as a cache replacement algorithm
  - Popular media streams are excessively cached in the network
  - Unpopular media streams disappear from the network

| | | | |
|---|---|---|---|
| popularity | 4 | 2 | 1 |
| amount | 5 | 2 | 0 |

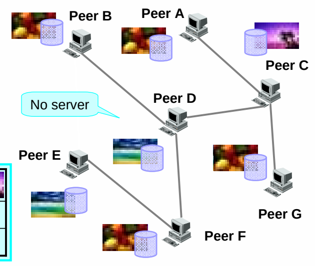





## Solution
## - supply-demand based cache replacement algorithm -

- Cache replacement algorithm considering balance between supply and demand for media streams in the network
  1. Estimation of supply and demand per media stream
  2. Determination of an appropriate media stream to be replaced (victim)
  3. Replacement the victim in a block-by-block basis

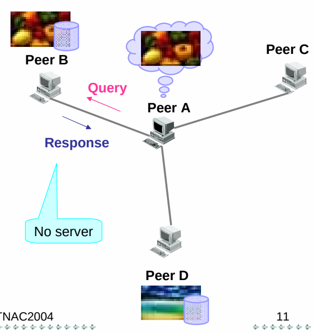
| | | | |
|---|---|---|---|
| popularity | 4 | 2 | 1 |
| amount | 4 | 2 | 1 |






## Step1: Estimation of Supply and Demand

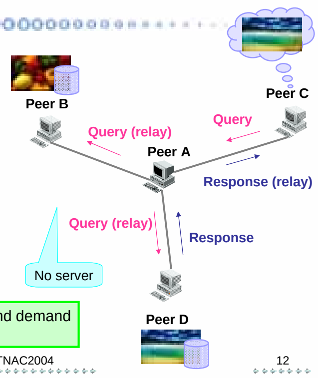
- Estimate supply and demand based on locally available information per round
  - Supply
    - Response message
  - Demand
    - Query message






## Step1: Estimation of Supply and Demand

- Estimate supply and demand based on locally available information per round
  - Supply
    - Response message
    - Relayed response message
  - Demand
    - Query message
    - Relayed query message




Every peer estimates supply and demand per media stream by itself

## Step2: Determination of Victim
### - bio-inspired approach -

- Our proposed media streaming is a distributed system
- In biology, insects, such as ants, also construct a highly structured organization only through indirect communications among individuals
  - Recently proposed model of division of labor in a colony of wasps can be transformed into a decentralized adaptive algorithm of task allocation [12]

⇓

- By regarding replacement as task, "division of labor and task allocation" can be applicable

[12] E.Bonabeau, M. Dorigo, and G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems," Oxford University Press, 1999



## Step2: Determination of Victim
### - division of labor and task allocation -

- Original model is division of labor in a colony of social insects
  - There are a lot of Individuals and several types of tasks in a colony
  - It is preferred that the number of individuals that perform a task follows the demand to the task

⇓

- Probability $P_{ji}$ that individual $j$ performs task $i$: $P_{ji} = \frac{s_{ji}^2}{s_{ji}^2 + \theta_{ji}^2}$
  - $s_{ji}$ : stimulus (demand) to task $i$ observed at $j$
  - $\theta_{ji}$: individual $j$'s response threshold to task $i$

$$\theta_{ji} = \begin{cases} \theta_{ji} - \varepsilon & \text{if } j \text{ performed task } i \\ \theta_{ji} + \varphi & \text{if } j \text{ did not performed task } i \end{cases}$$

⇒ •Once performing a task, the individual tends to devote itself to the task
•Otherwise, it tends to hesitate to perform the task



## Step2: Determination of Victim
### - applying the model to cache replacement -

- Original division of labor and task allocation

$$P_{ji} = \frac{s_{ji}^2}{s_{ji}^2 + \theta_{ji}^2}$$

$i$ : task
$j$ : individual
$P_{ji}$ : probability $j$ performs $i$
$s_{ji}$ : stimulus to $i$ observed at $j$
$\theta_{ji}$ : $j$'s response threshold to $i$

⇒

- Proposed cache replacement algorithm

$$P_r(j,i) = \frac{s^2(j,i)}{s^2(j,i) + \theta^2(j,i)}$$

$i$ : media stream
$j$ : peer
$P(j,i)$ : probability $j$ discards $i$
$s(j,i)$ : $\frac{\text{supply}}{\text{demand}}$ of $i$ observed at $j$
$\theta(j,i)$ : $j$'s response threshold to $i$

As a media excessively exists, it tends to be discarded

⇒ Supply and demand for media streams will be well-balanced



## Step2: Determination of Victim
### - extension on the equation -

- Ratio $P_r(i)$ that media stream $i$ is replaced

$$P_r(j,i) = \frac{s^2(j,i)}{s^2(j,i) + \theta^2(j,i) + l^2(j,i)}$$

  - $l(j,i)$: ratio of the number of locally cached blocks to the number of blocks in media stream $i$
    - To restrain the replacement of fully or well-cached streams
- A victim is chosen with probability $\frac{P_r(j,i)}{\sum_{i \in M-m} P_r(j,i)}$
  - $M$: set of cached media streams
  - $m$: stream being watched



## Step3: Replacement of Victim in Block-by-Block Basis

- Discard the victim in a block-by-block basis from the head or tail of the media stream at random
- Regulate the response threshold as follows

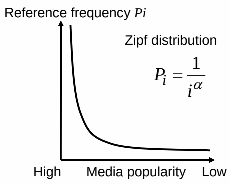
$$\theta(j,i) = \begin{cases} \theta(j,i) - \varepsilon & \text{if } i \text{ is victim} \\ \theta(j,i) + \varphi & \text{if } i \text{ is not victim} \end{cases}$$

  - Media $i$ is to be discarded more often once it is chosen as a victim



## Simulation Model

- Random network with 100 peers
- 40 media streams whose popularity follows a Zipf-like distribution with $\alpha$=1.0
- The inter-arrival time between two successive requests for the first media stream follows the exponential distribution whose average is 20 minutes



# Evaluation Criteria

- Continuity of media play-out
  - Completeness of block retrieval

$$\text{Completeness} = \frac{\text{Number of retrieved blocks in time}}{\text{Number of blocks in a media stream}}$$

- Adaptability to changes in media popularity
- Sensitivity to parameter settings of cache replacement algorithm

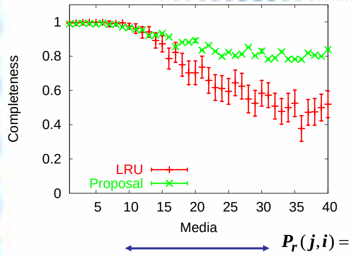
$$P_r(j,i) = \frac{s^2(j,i)}{s^2(j,i)+\theta^2(j,i)+l^2(j,i)}$$

$$\theta(j,i) = \begin{cases} \theta(j,i) - \varepsilon & \text{if victim} \\ \theta(j,i) + \varphi & \text{if not victim} \end{cases}$$



# Continuity of Media Play-out

- LRU vs. proposal -




Parameter settings

| | |
|---|---|
| θ(j,i): default | 0.5 |
| θ(j,i): range | 0.001~1 |
| ε | 0.01 |
| φ | 0.001 |

High ← Popularity → Low

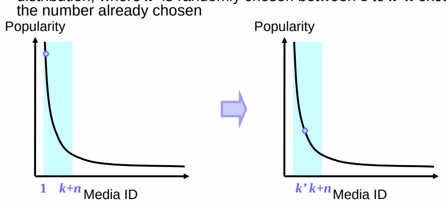
$$P_r(j,i) = \frac{s^2(j,i)}{s^2(j,i)+\theta^2(j,i)+l^2(j,i)}$$

$$\theta(j,i) = \begin{cases} \theta(j,i) - \varepsilon & \text{if victim} \\ \theta(j,i) + \varphi & \text{if not victim} \end{cases}$$



# Media Popularity Change Model

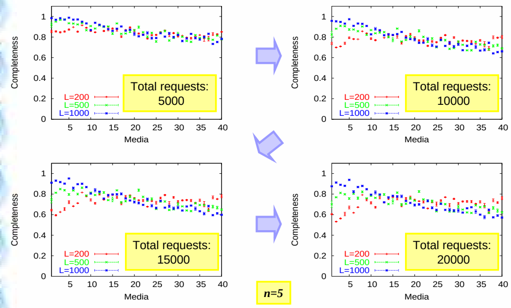
- Popularity changes every $L$ media requests
- When the change occurs, the Zipf-like distribution is renewed to a new one as follows
  - $k$ th popular media stream becomes $k'$ th popular in the next Zipf-like distribution, where $k'$ is randomly chosen between **1** to $k+n$ except for the number already chosen




[16] Kun-Lung Wu, Philip S. Yu, and Joel L. Wolf. Segment-based proxy caching of multimedia streams. In *Proceedings of the 10th International WWW Conference*, pages 36-44, 2001.



# Adaptability to Changes in Media Popularity

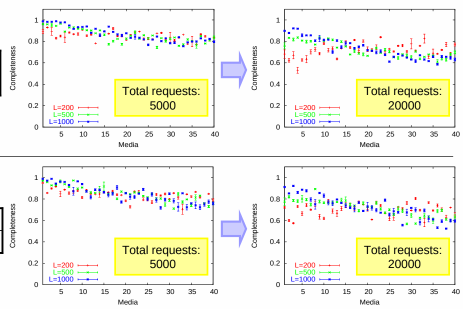





# Sensitivity to Parameter Settings of Cache Replacement Algorithm

| | |
|---|---|
| ε | 0.1 |
| φ | 0.01 |

| | |
|---|---|
| ε | 0.001 |
| φ | 0.0001 |






# Conclusion and Future Work

- Conclusion
  - We discuss the media streaming on pure P2P networks
  - We propose a supply-demand based cache replacement algorithm
    - Continuous media play-out is accomplished independent of media popularity
    - Proposed algorithm flexibly adapts to changes in media popularity without any parameter adjustment
- Future Work
  - We evaluate proposed mechanisms in more realistic situations where a peer randomly joins and leaves our system

# Thank You