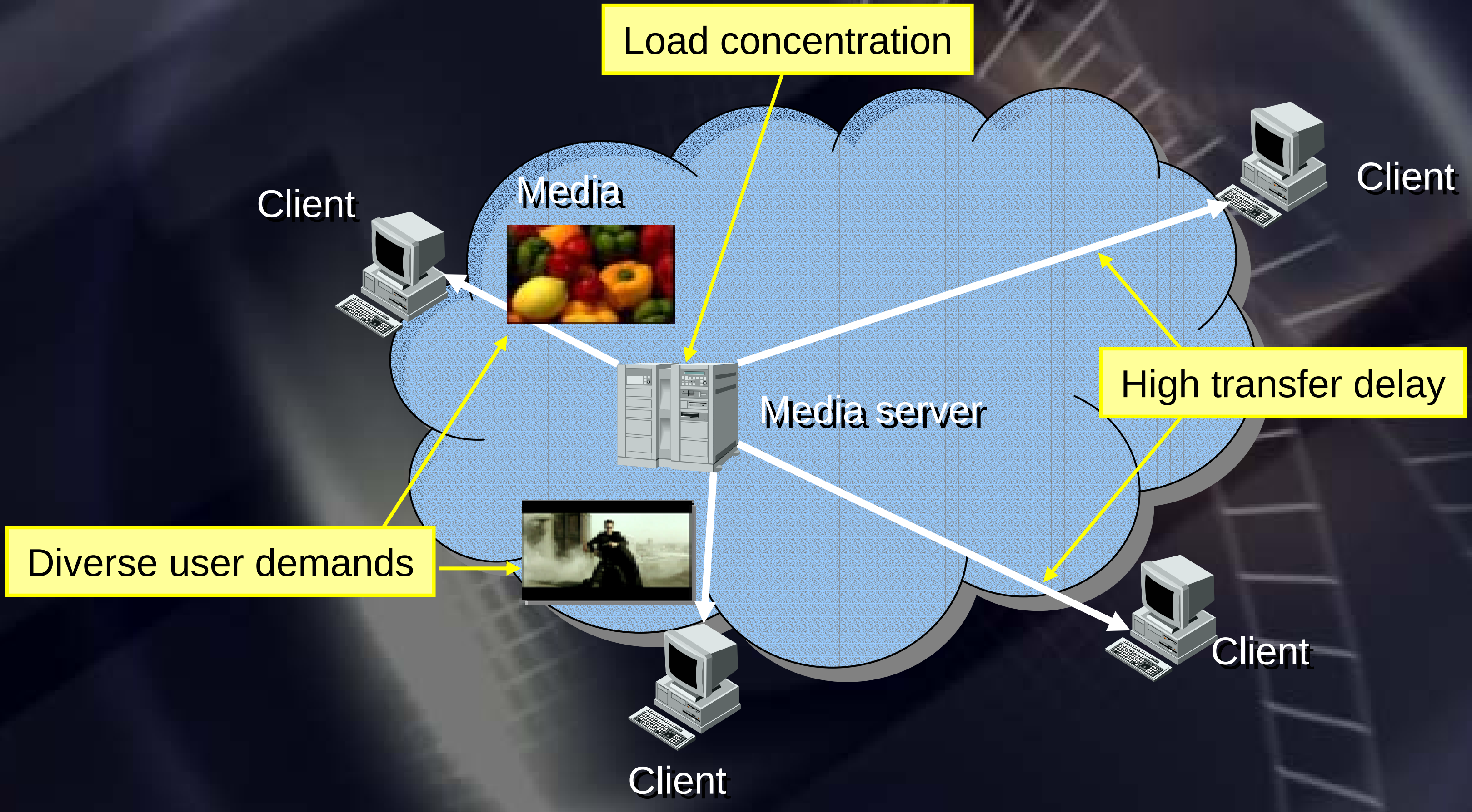
Scalable and Continuous Media Streaming on Peer-to-Peer Networks

Masahiro Sasabe
Osaka University, Japan
m-sasabe@ist.osaka-u.ac.jp

Content

- Backgrounds
- Media Streaming on pure Peer-to-Peer Networks
- Research Targets
- Outline of Proposed Mechanisms
  - Scalable Search Methods
  - Block Retrieval Methods
- Simulation Results
- Conclusion and Future Work

Backgrounds
- server-client model -



P2P is a network paradigm to solve these problems

Backgrounds

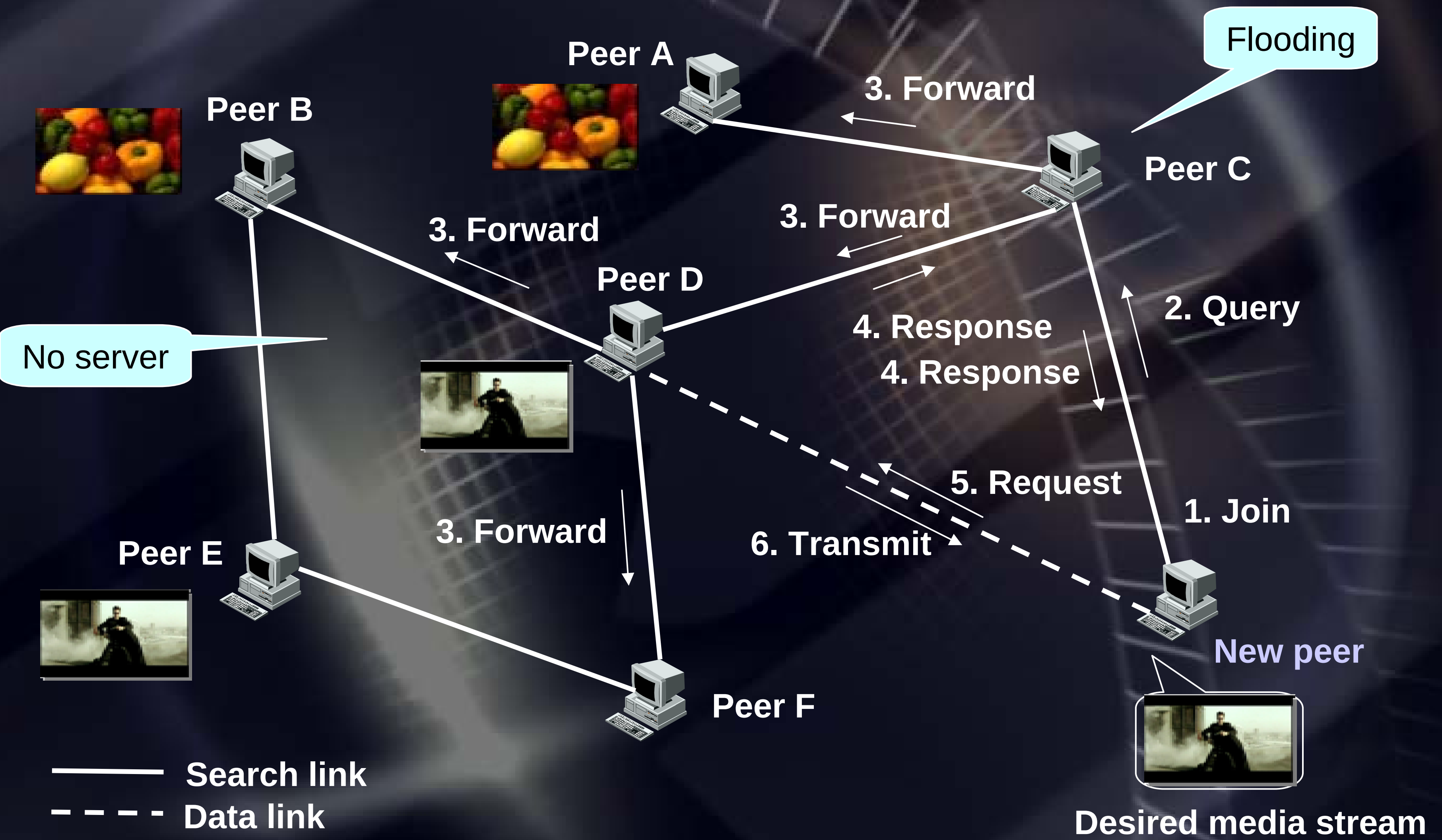
- related works on P2P media streaming -

- Most of research works on P2P media streaming construct an application-level multicast tree
  - Effective for live-media streaming
  - Not effective for on-demand media streaming
  - The root of the tree is a single point of failure

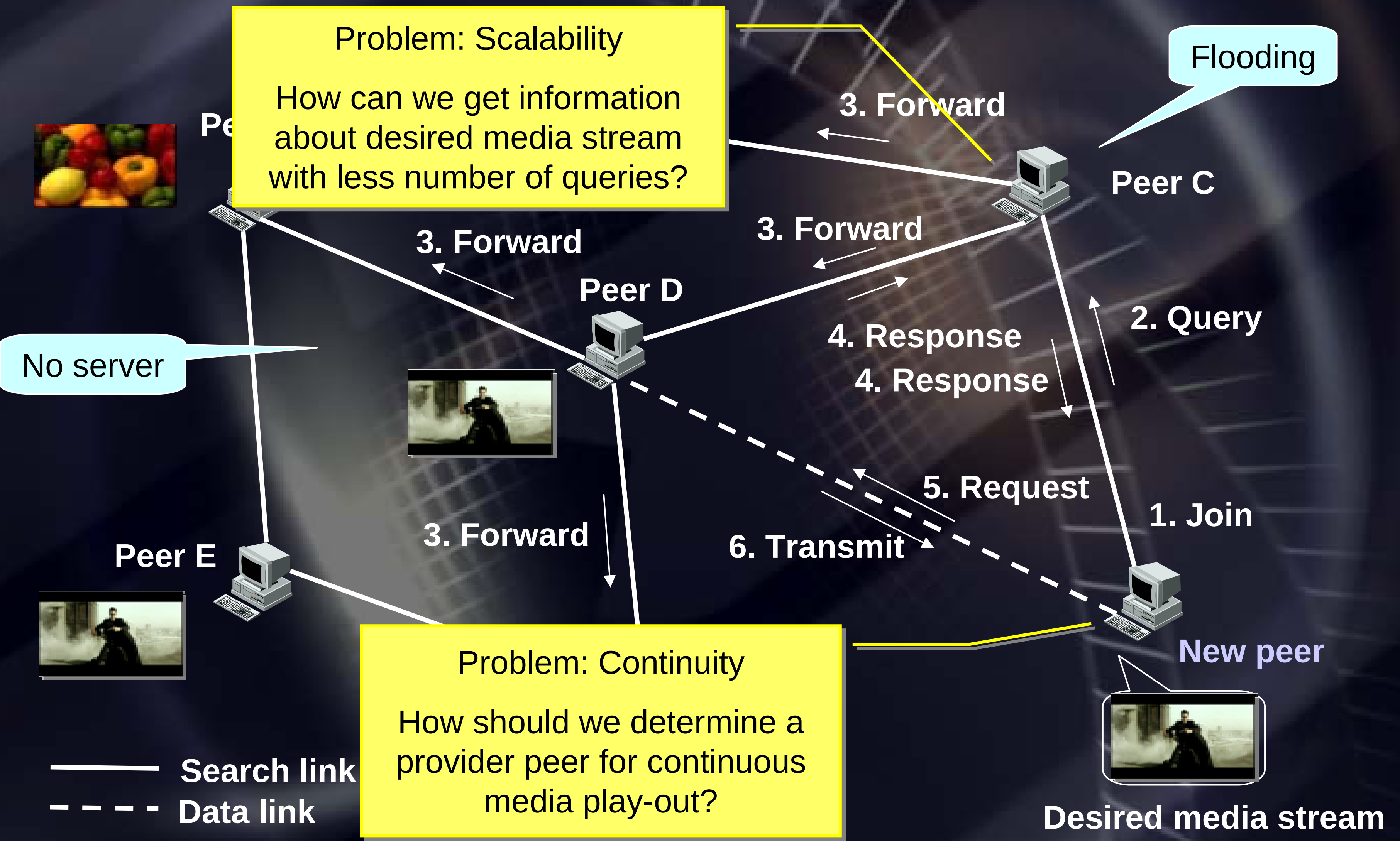
⇓

- We discuss effective methods for on-demand media streaming on pure P2P networks

Media Streaming on pure P2P Networks



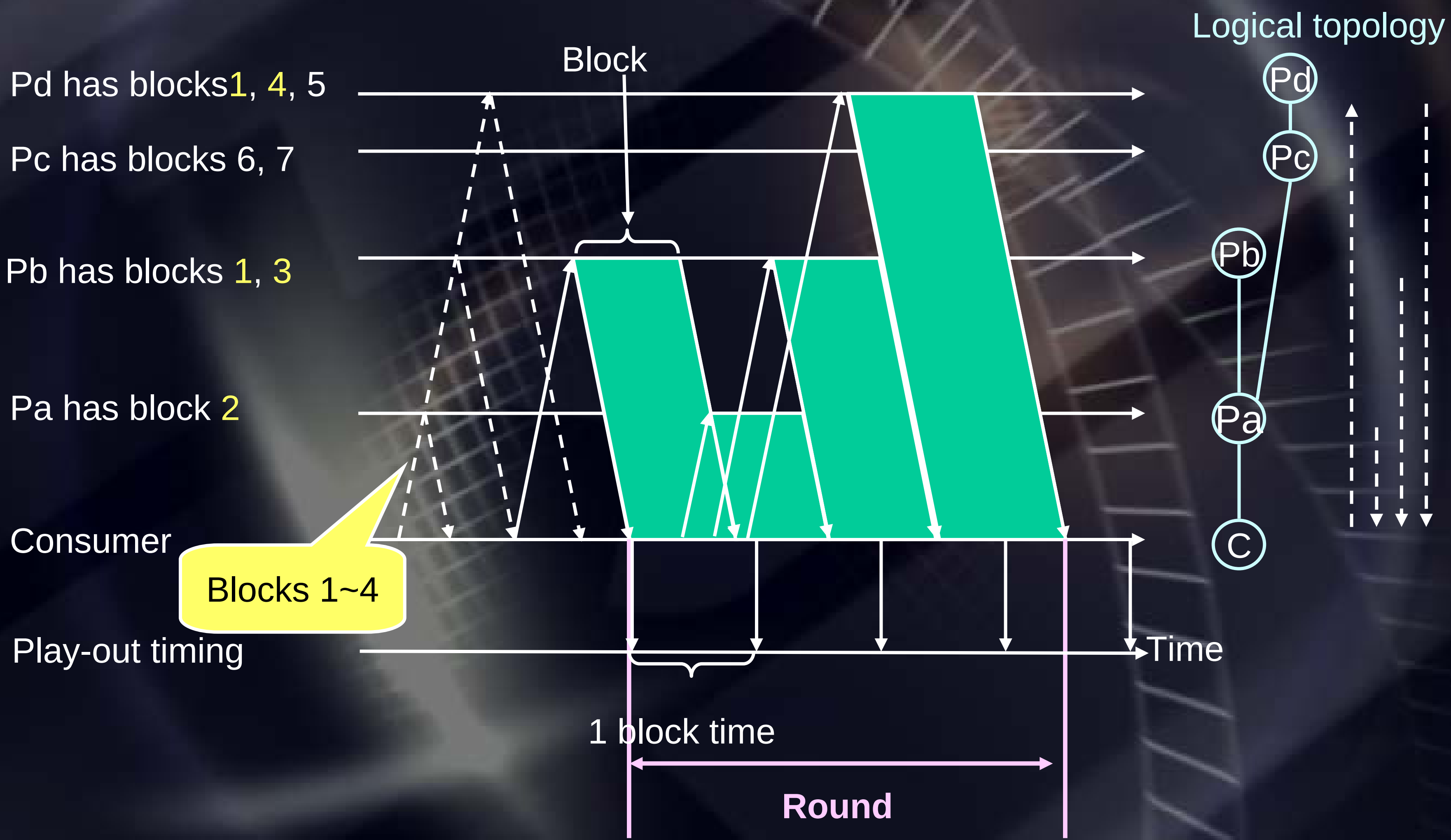
Media Streaming on pure P2P Networks



Research Targets

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

Outline of Proposed Mechanisms
- per-group block search and retrieval -



Advantage of Per-group Based Search

- There is a temporal order of reference in a media stream
  - A user watches a media stream from the beginning to the end

↓

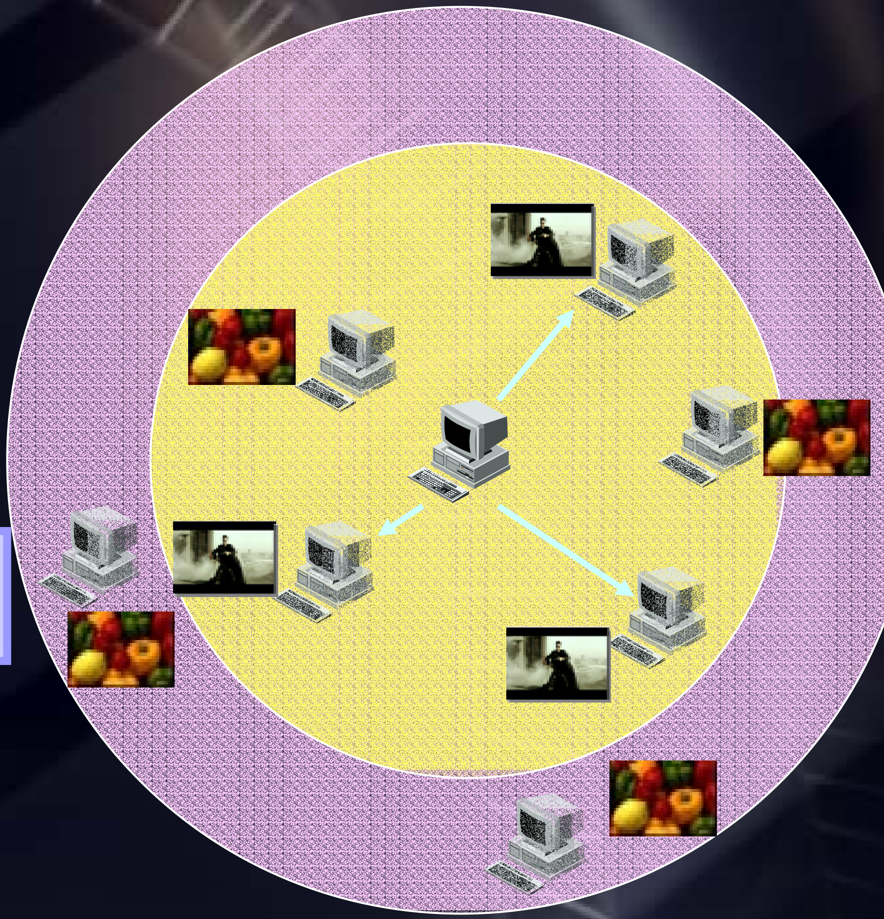
- We can expect that a peer, which replies a response message in the current round, has some blocks of the next round
  - A peer can estimate candidates of provider for blocks of the next round from the search results in the current round

Scalable Search Methods

- 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

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



Desired media stream

Provider Peer Determination Algorithm

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

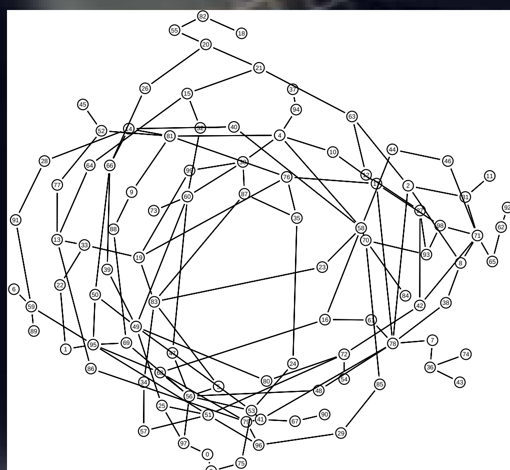
1. Calculate a set of peers from which it can retrieve a block in time.
2. Select a peer from the set
   a. Select a peer whose estimated retrieval time is the smallest among peers in the set (SF Method)
   b. Select a peer with the lowest possibility of block disappearance among peers in the set (SR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

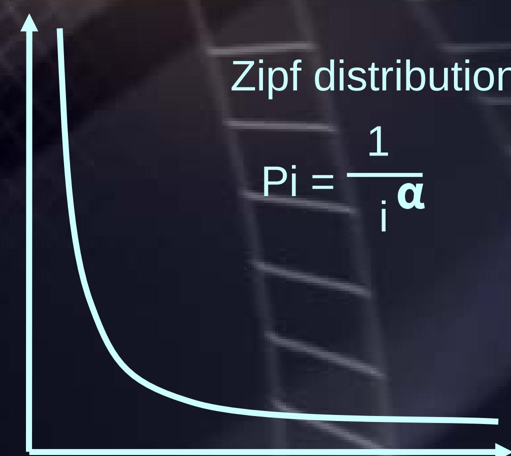
Provider Peer Determination Algorithm (detail)

1. $j \leftarrow r$
2. Calculate set *S*, a set of peers having block *j*
   a. if $S = \phi$
      $Tf(j) \leftarrow Tp(j)$, $j \leftarrow j+1$, repeat Step2
   b. if $S \neq \phi$ go to Step3
3. Derive set *S*', a set of peers from which a provider peer can retrieve block *j* by deadline *Tp(j)*
   $max(Tf(j\text{-}1),$
   $Tnow+R(i))+B(j)/A(i) <= Tp(j)$
   a. if $S' \neq \phi$ go to Step4
   b. if $S' = \phi$
      $Tf(j) \leftarrow Tp(j)$, $j \leftarrow j+1$,
      go back to Step2
4. Determine provider peer *P(j)* from *S'*
   - SF (Select Fastest) Method
   - SR (Select Reliable) Method
5. Derive the estimated completion time of retrieval *Tf(j)* and the time *Tr(i)* to send a request for block *j*
   $Tf(j) = max(Tf(j\text{-}1), Tnow+R(i))$
   $+B(j)/A(P(i))$
   $Tr(j) = Tf(j) – R(P(j)) - B(j)/A(P(i))$
6. *jf j = kN, finish*
   otherwise $j \leftarrow j + 1$,
   go back to Step2

Simulation Model

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





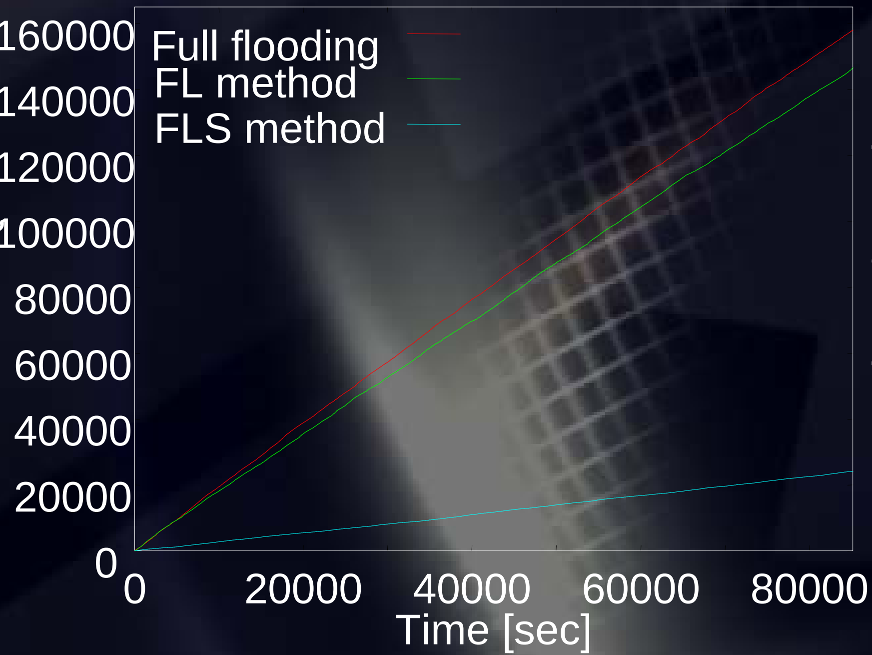
Evaluation Criteria

- System scalability
  - Average number of queries that a peer receives during the simulation
- Continuity of media play-out
  - Completeness of block retrieval

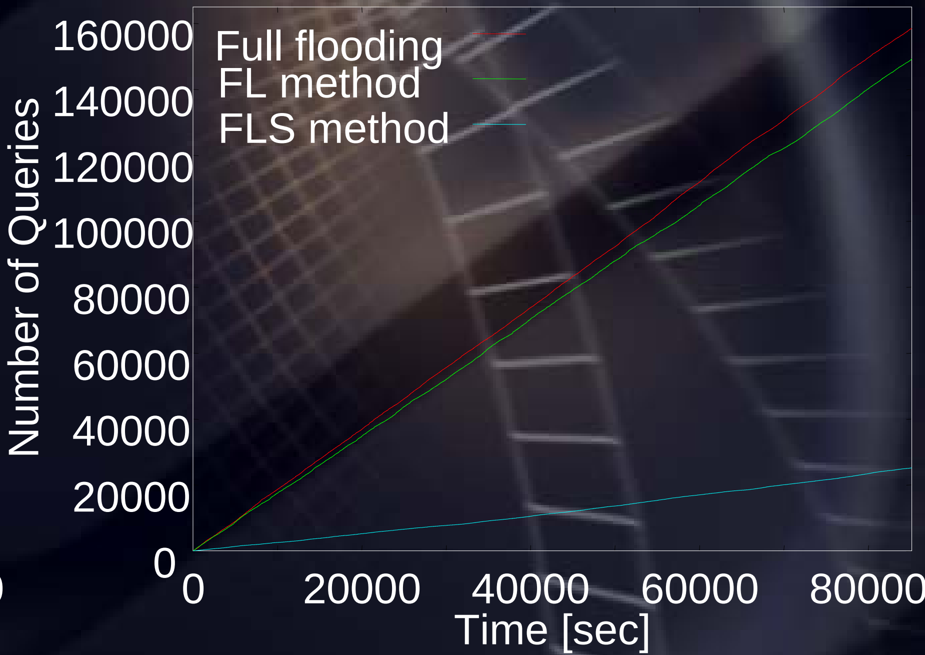
$$\text{Completeness} = \frac{\text{Number of retrieved blocks in time}}{\text{Number of blocks in a media stream}}$$

Simulation Results
- number of queries -

SF method

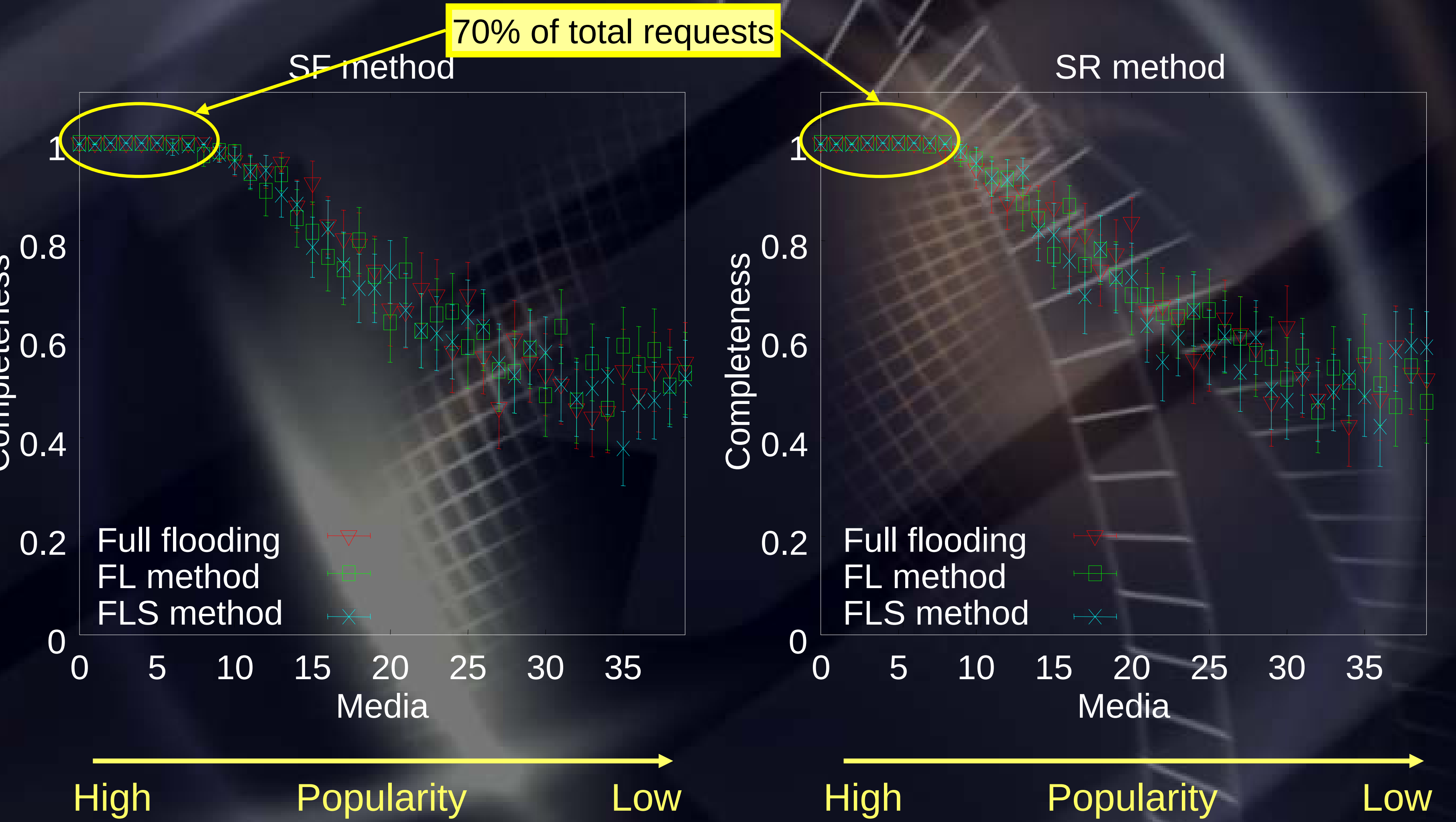


SR method



Simulation Results

- completeness with 95% confidence interval -

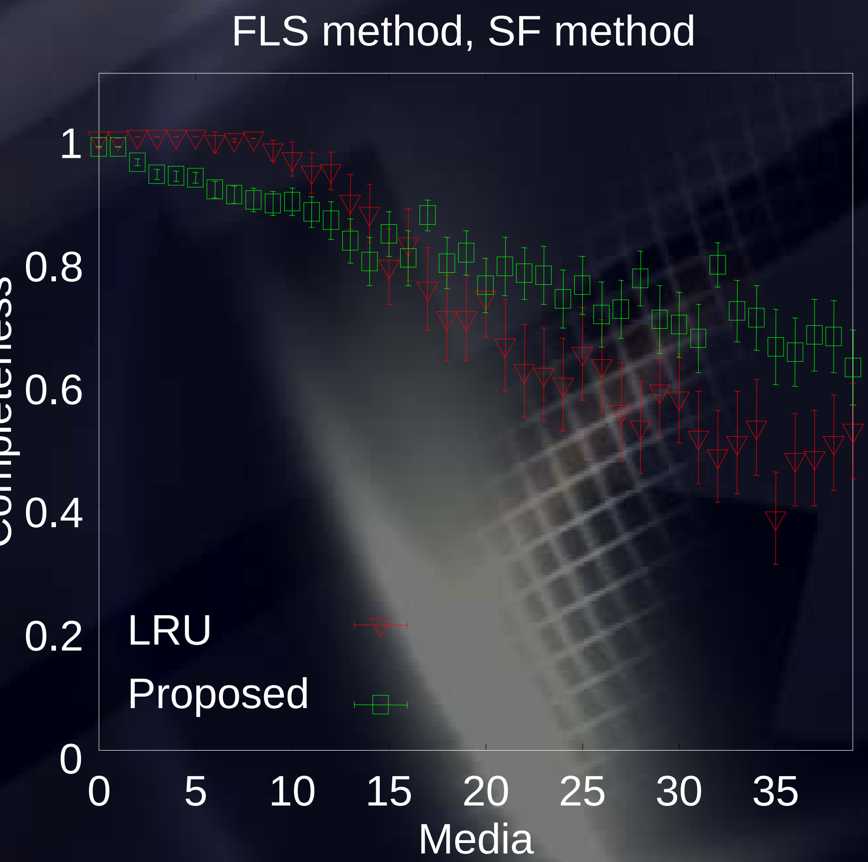


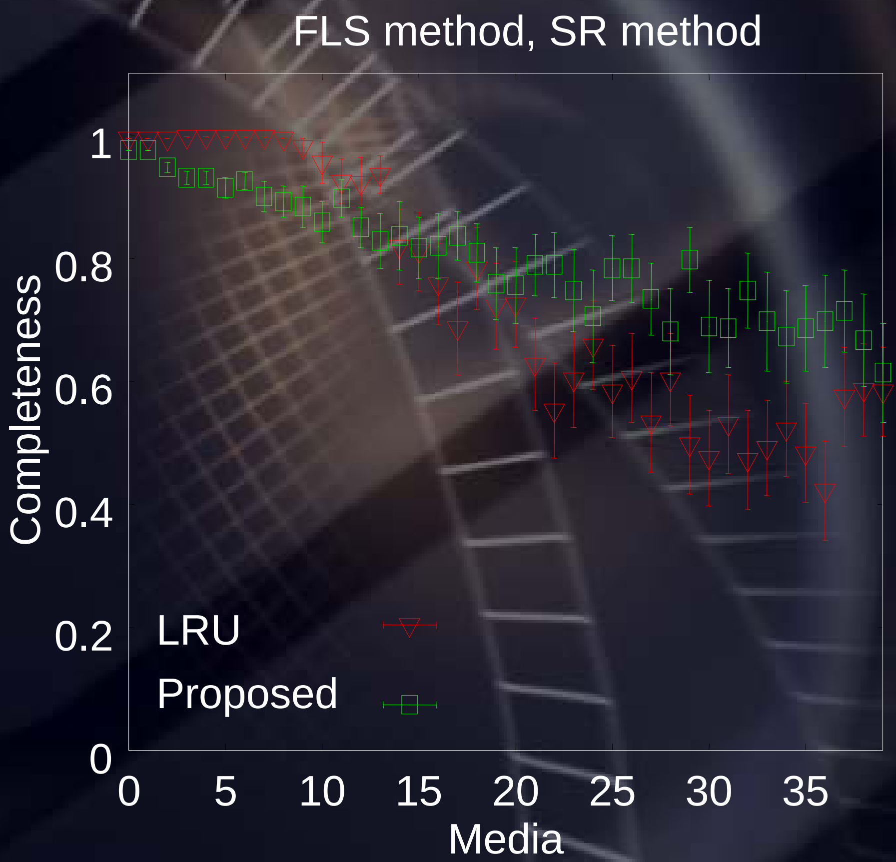
Conclusion and Future Work

- Conclusion
  - We discuss the media streaming on pure P2P networks.
  - We proposed block search and retrieval methods for continuous and scalable media streaming.
  - Through simulation experiments, we have shown that the FLS method can provide users with continuous media play-out without introducing extra load on the system for popular media streams.
- Future Work
  - Improve the completeness of unpopular media streams
    - We are now considering an effective cache replacement algorithm, which takes into account supply and demand for media streams.
  - Evaluate proposed mechanisms in more realistic situations where network conditions and peer locations dynamically change.

Appendix: Simulation Results

- LRU v.s. proposed cache replacement algorithm -





High Popularity Low

High Popularity Low