Stir: Spontaneous Social Peer-to-Peer Streaming

Anh Tuan Nguyen (Dept. of Informatics, Uni. of Oslo)
Baochun Li (Dept. of Electrical and Computer Engineering, Uni. Of Toronto)
Michael Welzl (Dept. of Informatics, Uni. of Oslo)
Frank Eliassen (Dept. of Informatics, Uni. of Oslo)

High churn rate is a very challenging problem in live P2P streaming

- Approaches
  - Peer dynamic models, e.g., Kumar *et. al.* [1].
  - Analyzing traces, e.g., Vu *et. al.* [2].
  - Enhanced coding techniques, e.g., Wang and Li [3]

- Not much attention has been paid on the origin of the problem
  - The interest of users in the streaming session: users leave the session because they are not interested in the content.

[1] R. Kumar, Y. Liu, and K. Ross, “Stochastic Fluid Theory for P2P Streaming Systems,” in Proc. of IEEE INFOCOM, 2007.

[2] L. Vu, I. Gupta, K. Nahrstedt, and J. Liang, “Understanding the Overlay Characteristics of a Large-scale Peer-to-Peer IPTV System,” ACM Trans. on Mult. Computing, Communications and Applications, Feb 2011.

[3] M. Wang and B. Li, “R2: Random Push with Random Network Codingi n Live Peer-to-Peer Streaming,” IEEE Journal on Selected Areas in Communications, vol. 25, no. 9, pp. 1655–1666, Dec. 2007.

Our Approach: Social Networking

- Social networking helps to know user interests: how are they interested in a particular session?
  - Recognize stable peers
  - Connect peers with a similar level of interest
  - > mitigate the impact of peer dynamic

- Offers social activities to users
  - Hypothesis: the more socially active a user is, the longer she stays in the session
    - *Homophily:* the tendency of people with similar characteristics to be connected [4].
    - There is a correlation from social networks to user behavior on the Web [5]: the more time they spend talking, the stronger the relationship is.

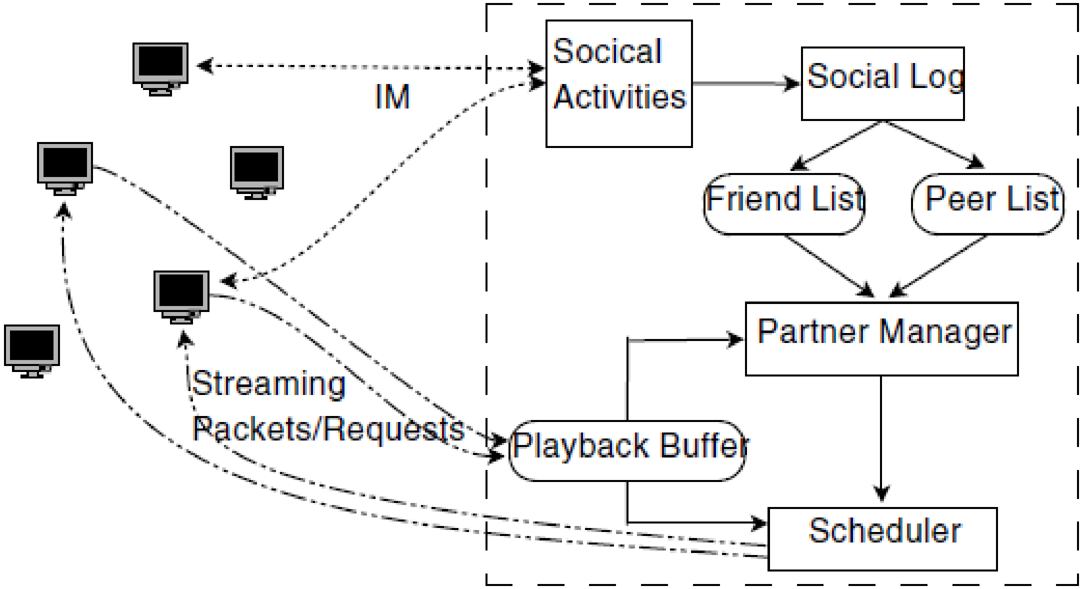
[4] M. McPherson, L. Smith-Lovin, and J. M. Cook, “Birds of a Feather: Homophily in Social Networks,” *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[5] P. Singla and M. Richardson, “Yes, There is a Correlation - From Social Networks to Personal Behavior on the Web,” in *Proc. of ACM WWW*, 2008, pp. 655–664.

Stir: a multi-channel live P2P streaming system

- Each Stir user U has a profile.
- If authenticated, when U selects a channel, the server will send an IP address list of some available peers in the channel to U.
- Friendships are to be established on-the-fly, as a group of users watch the same channel.
- U can post comments to one of the Stir online forums. This forum is only visible to those who are watching the same channel with U.
- U can have a private chat with another user V. U can add V to her friend list.

Stir: a pull-based P2P streaming protocol



The architectural design of Stir, with an emphasis on the tight integration between streaming quality and spontaneous social network relationships

- Each Stir peer has two lists for potential partners
  - The neighbor list is updated by a gossip-based mechanism
  - The friend list contains friends who are also watching the same channel.

Choosing partners: relying on social capacity or network capacity?

- Utility function: peer F calculates utility of peer Q

$$U_Q = (1 - \alpha - \beta)*C(Q) + \beta*S(Q) + \alpha*F(P,Q)$$

  - *C:* a capacity-related function, S: a social-related function, and F: is a friendship-related function.
  - β: social coefficient, α: friendship coefficient

- In traditional non-social P2P streaming
  - α and β are 0 as social factors do not exist.
- In previous social-based P2P systems (for file-sharing)
  - α is close to 1 as they discourage connections with strangers.

U_Q = (1 – α – β)*C(Q) + β*S(Q) + α*F(P,Q)

- In this paper
  - $C(Q) = \frac{B_Q}{max_{i \in L} B_i}$ (B: bandwidth)
  - $S(Q) = \frac{N_Q}{max_{i \in L} N_i}$ (N: number of friends)
  - $F(P, Q) = \begin{cases} 1 & if\ P\ and\ Q\ are\ friends \\ 0 & otherwise \end{cases}$

- Complicated functions, including more metrics, can be used. E.g.:
  - RTT can be added to C(Q)
  - Social data Q exchanges can be added to S(Q)

Partner Manager

- Sending partner requests
  - calculates utility values of peers in the neighbor list and the friend list.
  - sends partner requests to a certain number of peers, which has the highest utility values
  - if a candidate Q accepts the request, P adds Q to its partner list, which is used for buffer map exchanges and video data requests.

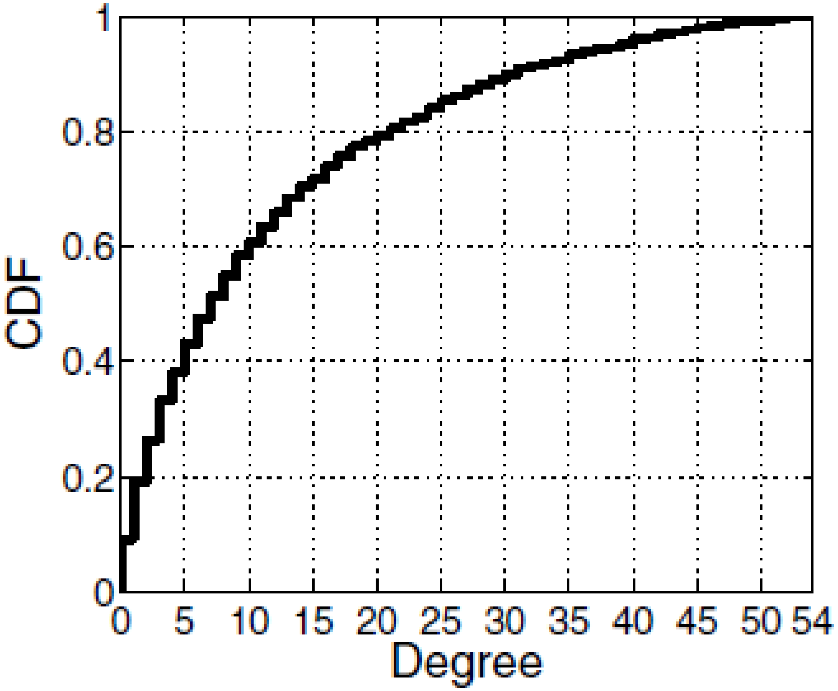
- Accepting requests
  - always accepts friends (a design principle)
  - otherwise, based on the utility function

Evaluation

- Own discrete-event flow-based simulator
- Data preparation
  - Social graph: from Flickr
    - All members of a Flickr group are considered as users of a streaming session.
    - If a member P is in the contact list of another member Q, they are friends of each other in the session. Since we consider relationships among users who join the same session, non-members in the contact lists of the members are removed.
  - Peer dynamics
    - Real snapshot trace of PPLive
    - Three scenarios: low, medium, and high dynamic

Photo Computer Art group

- 1280 members
- ~10% of the members have no friends,
- ~80% have fewer than 20 friends,
- and the other 20% have from 20 to 54 friends

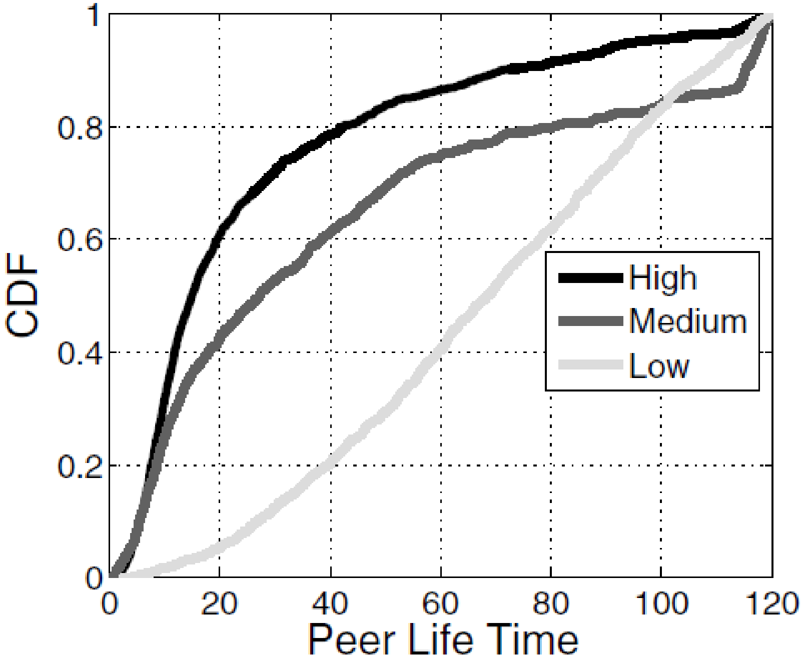


Friendships among peers

Peer dynamics

- The high dynamic scenario has up to 60% of peers staying in the session less than 20 minutes and ~ 90% staying less than one hour.

- Up to 60% of the peers stay longer than one hour in the low dynamic case.



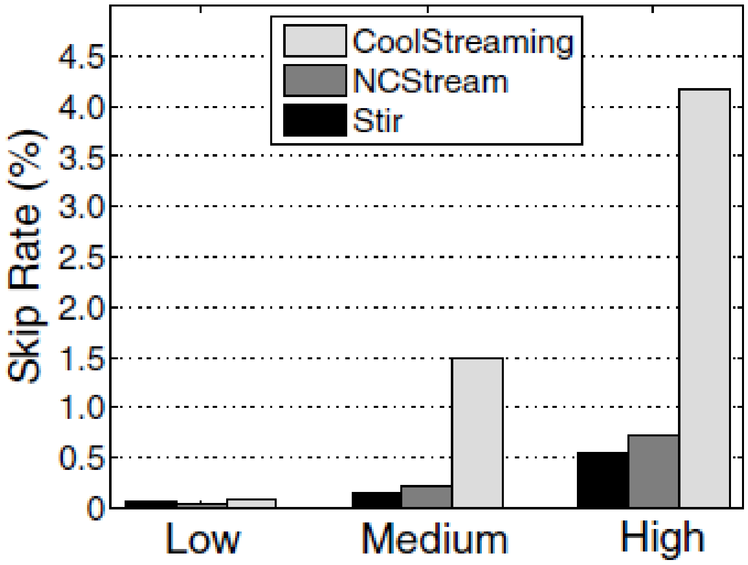
Peer churn

Comparison

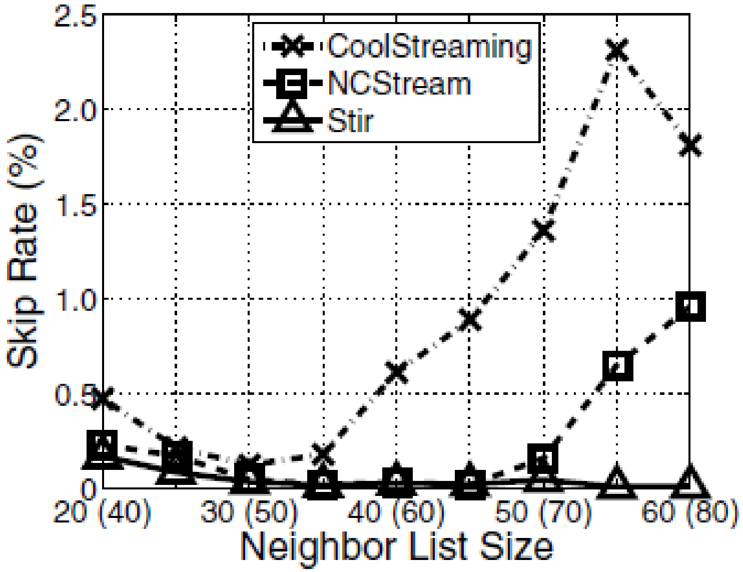
- Related work
  - CoolStreaming-like protocol (named CoolStreaming)
  - Network coding based protocol (named NCStreaming)

- Simulation setting
  - The server is set to be able to serve ~5% of users
  - Streaming rate: 400 Kbps
  - Bandwidth capacity (download, upload): (450, 300), (550, 450), (700, 650), (750, 700) Kbps

Stir offers better skip rates with a small neighbor list



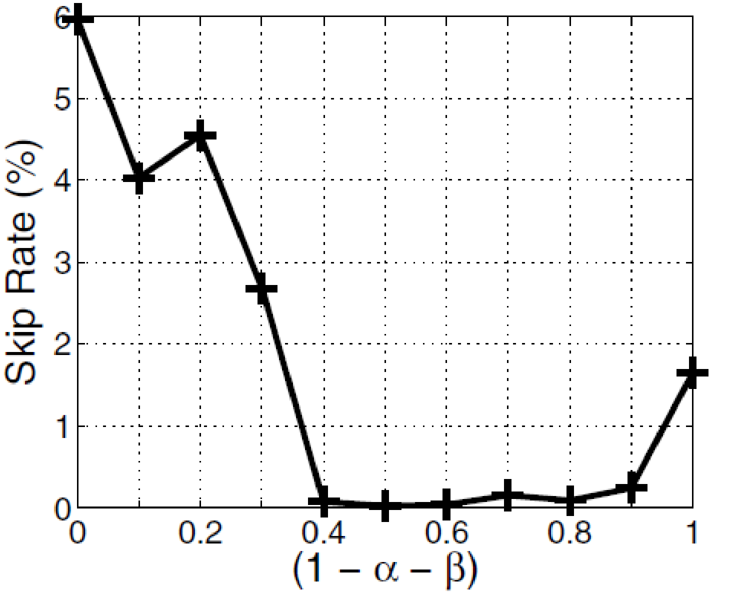
Skip rate with different peer churn patterns



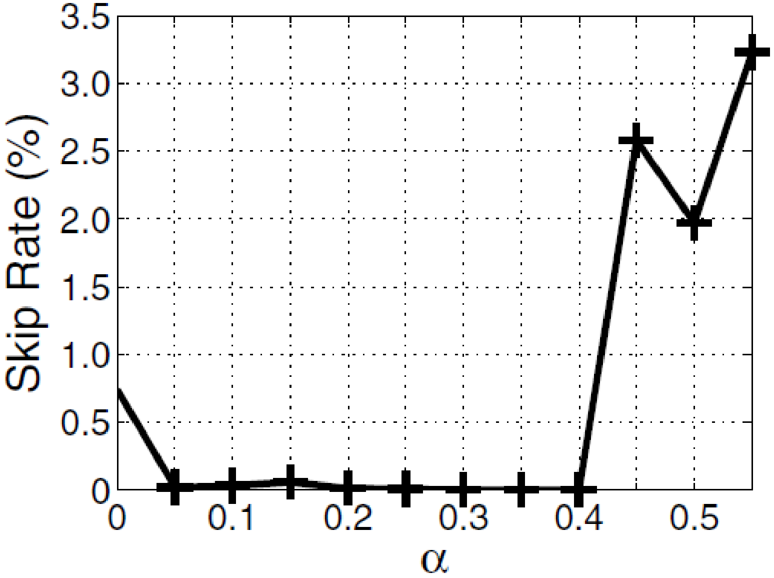
Skip rate with different neighbor list sizes (The numbers in parentheses denote the size of the neighbor list in CoolStreaming and NCStream)

- Reason: have better connections from the friend list.

Insights of Stir



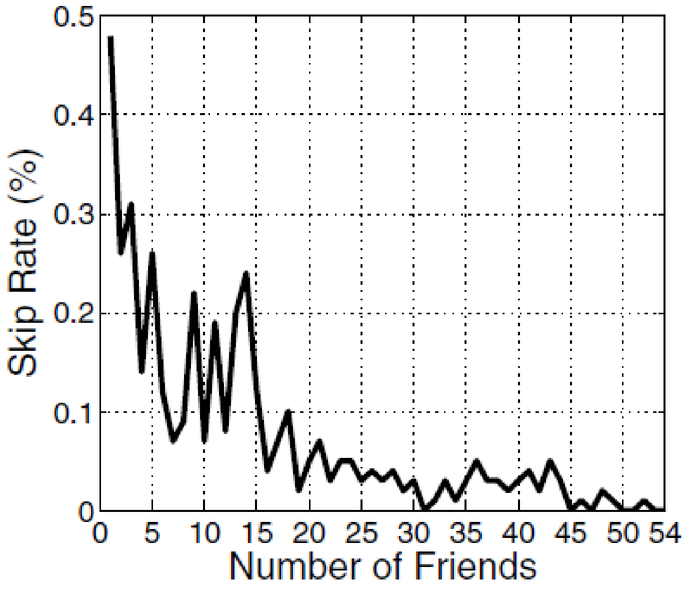
Network capacity Vs. Social factors



Social capacity Vs. Friendship

- High network capacity does not imply high social capacity
- Peers with high social capacities have more friends to serve, and being served. Its friends may not have sufficient bandwidth
- Network capacity, social capacity and friendship have their own roles in the partner selection process

Values of being famous in Stir



Skip rate on different number of friends

- The more friends you have, the better the quality you receive.
- A rational peer will make a large number of cheap 'spontaneous' friends to improve its quality?
  - This is a common issue of social-based applications, e.g. in Facebook games.
  - One possible solution: use some requirements in the friend making process such as friendship can only be formed after two users sometimes chat to each other.

Conclusion

- Stir is the first attempt to build up a social P2P live streaming system
  - With spontaneous social networking, i.e. social relationships are formed on-the-fly, more entertaining streaming services can be achieved. E.g., users watch football game online `together’.
  - The system benefits from users’ social activities: streaming quality is improved.

Questions?

Thank you