InstantLeap: An Architecture for Fast Neighbor Discovery in Large-Scale P2P VoD Streaming

Xuanjia Qiu[1], Wei Huang[1], Chuan Wu[1], Francis C.M. Lau[1], Xiaola Lin[2]

[1]Department of Computer Science, The University of Hong Kong, Hong Kong,{xjqiu,whuang,cwu,fcmlau}@cs.hku.hk

[2]Department of Computer Science, Sun Yat-Sen University, P. R. China, linxl@mail.sysu.edu.cn

I. DETAILED COMMENTS CORRESPONDING TO THE REVIEWS

We appreciate the reviewers valuable comments and suggestions. We have carefully revised the paper accordingly, and explain the revision details below.

A. Reviewer 1's Comments

1. "I think it is important that the authors discuss their work with respect to RanSub by Dejan Kostic et al. (see USENIX 2003 and ICDCS 2004) which first established some of the fundamental ideas about using random subsets as a mechanism to support scalability. I know that there are significant differences between the RanSub application domain and what the authors are trying to do here, but it is an important piece of early work that deserves mention since it is some of the earliest and best work on the idea of load-balancing via random selection."

We have added the discussion on the relationship between our work and RanSub in the section of "related work".

2. "the use of the word 'size' starting in Lemma 1 and running throughout the analysis is a bit confusing. I think the authors really mean "span" in that what they are referring to is the percentage of the m groups that a peer has shortcut neighbors to. When I read 'size', I thought they were referring to

the total number of short cut peers, some of whom will be in the same group for redundancy purposes, so the analysis didn't make sense. It took me quite a while to understand the analysis until I had that insight, so a bit of care at the beginning to explain exactly what you mean by 'size', or by using the word "span" instead, could be helpful."

We appreciate that the reviewer has pointed it out. We have revised Lemma 1 accordingly, and touched up other parts in Section V for consistent expression.

3. "the section III.B just didn't make any sense to me at all. I really didn't understand what point the authors were trying to make with the "condensed" graph. I would suggest dropping it since it doesn't seem to be needed to understand the rest of the paper."

B. Reviewer 3's Comments

1. "The overhead of maintenance seems quite extensive. Authors did not mention about the overhead of keeping neighbor lists fresh. Mainly, each peer in the list of neighbors should be probed periodically, to make sure its still alive, calculate its N_u (set of neighbors the peer uploading to) and playback time or group information. Without such a periodical probing SNM condition cannot accurately be computed (while the authors just proposed the idea of SNM without evaluating that!). I think having connection to 0.63*m*redundancy peers for the shortcut list in addition to the 30-50 streaming neighbors is quite alot. The authors just showed the overhead of peer join/leap and departure and didnot show the overhead of maintining this many connections fresh in terms of their status."

In fact, we mention the periodical exchanges of neighbor lists among peers in Section VI, to get the updated information of each neighbor. per 60 seconds – Althouth the number 0.63*m*redundancy is not small, but these shortcut neighbors only need loosely maintained, compared with the streaming neighbors which need tightly maintained. In the simulation part, we show the aggregate traffic overhead incurred by maintaining streaming neighbors and shortcut neighbors is acceptable in Fig. 6. The traffic overhead is lower than 2.5Kbps, equivalent to 0.5% of the streaming rate. We argue that it's worthwhile to trade some traffic overhead for improvement of user experence.

2. "SNM condition is mentioned in section IV.A but it hasnt been evaluated. Clearly SNM gives a

dynamic number of shortcut neighobrs from each group, however the evaluated result in Section VI.E.2, considers a static parameter for the redundancy. It is not clear for this reviewer how SNM can work and how much more benefit and overhead it incurs to the design."

a peer tries to find a number of peers (H at maximum) that can maximize the left side of the SNM condition within maximal L times of neighbor list exchanges.

Section VI.E.2 refers to a upper bound of the number of shortcut numbers per group. SNM is a stop criterion of the searching L is the steps of searching

We explain this in Section IV.

SNM also helps the balancing of reference times.

3. "For a journal paper, I think the proposed solution should be compared with existing central or DHT-based, DSL and the ring-based solutions. The performance decrease/increase in terms of time to find adequate neighbors, continuity index and overhead need to be evaluated."

doing

4. "It is not clear from the simulation description that how the block delivery and scheduling are simulated. Are the peers receving blocks from multiple neighbors of just one? Is the bandwidth availability/congestion considered in the transmission of the blocks to each peer or not? Is there an implicit assumption that peers in any group have all the corresponding blocks of that group? These are impotant as besides group membership, bandwidth availiblity and block availiablity are the determining factors in a smooth playback leap and continuity index of peers."

random mulitple no. no.

5. "Some of the evaluation parameters are not clearly described. The important factor on the performance of P2P streaming applications is the availability of bandwidth. It is not clear how much is the resource index in the evaluated scenario. If the setting is over provisioned (RI¿1), neighbor discovery becomes much more easier as any peer can be a potential supplying peer. On the other hand, if upload bandwidth is scarce, just having a neighbor or two in all groups doesnt provide any guarantee for smooth playback leap, as the peer not only should find a peer in the new group, but also should find a peer with adequate bandwidth in that group. I'd like to see the performance under scenarios with various bandwidth

distribution when RI is 1 or slightly more than 1."

for K=2, the resource index is 2. doing K=3(RI=1.5), K=4(RI=1) for the case of 5000 peers, K=2, delay=2 K=4, delay=5 Need data for (n=1000-10000, for K=3 and K=4) The experiment is running.

6. "One of the main issues with such a random neighbor lists, is the imbalance between upload bandwidth and the number of occurance of each peer in other peers' shortcut neighbor lists. Despite authors claim on Fig. 11 that "if a peer can allocate more upload bandwidth to each of its streaming neighbors, it tends to be known by more peers.", I cannot see such a claim on the mentioned figure. Ironically many peers with high bandwidth have their dots close to line y=0 (near 0 occurance)! In this case, load is not distributed proportional to the bandwidth of peers."

is it because of mixing peers from different groups together?

Expose the method of balancing the reference times. — searching procedure There are still problems with the current methods.