Using Coordinated Clients to Improve Live Media Contents Transmissions

Ronit Nossenson
Faculty of Computer Science
Jerusalem College of Technology
Jerusalem, Israel
ronit.nossenson@gmail.com

Omer Markowitz
School of Computer Science
The Interdisciplinary Center
Herzliya, Israel
markowitz.omer@post.idc.ac.il

***Abstract* — This research examines the possibility to significantly improve the quality of private live video transmission over the internet, as opposed to on-demand service, such as YouTube. To achieve this goal collaboration and coordination between small numbers of agents is carried out, using several communication methods such as wireless or cellular connections. Experimental performance results indicate that this method can significantly improve some performance parameters including packets jitter, with limited overhead.**

Keywords-Live Content Transmission; Multimedia QoS

I. INTRODUCTION

The increasing availability of various commercial products for private/domestic live video transmission on the one hand, and the many ways such a transmission could be received (PDA's, PC's, Smart Phones, Media Streamers, etc.) on the other, make it possible to exploit this medium faster than ever before. For instance, it is possible to broadcast a live video of a private family event to those unable to attend, an online transmission of a lecture, and as a matter of fact – experience almost any event without the need to physically being present "on location". All these emphasize the gap facing the poor quality of live video transmission that could be viewed today.

The commercial sector can afford purchasing the required bandwidth in order to transmit high quality video signals. However, this is not the case for the private domestic sector. An attempt to broadcast a live video signal on the internet often encounters difficulties, most of which arise from the inability to guaranty end-to-end QoS for the private individual, such as appropriate bandwidth or low bound on packet delay. For example, the common way to improve quality of viewing a video file located on a server (via services such as YouTube) is with the use of buffering on the viewer's computer. However, anyone who had experienced watching video over the internet surely noticed that this is generally not sufficient, in particular for live video streaming [4].

Another problematic issue arises from the inability to guaranty a sufficiently large bandwidth, which is mainly due to the competition over the bandwidth between clients (oversubscription factor). This could become even worse with longer broadcasting, even if theoretically the user's ISP provides the client with potentially enough bandwidth.

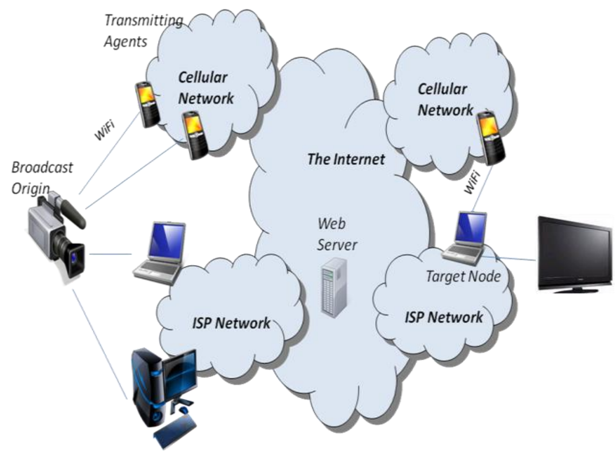

Figure 1: Coordinated agents uploading a private live video content.

Addressing the above issues usually aims at reducing the workload on the server and increasing the effective bandwidth. Most of the solutions are based upon Peer-to-Peer (P2P) or upon multicast at the network infrastructure level [5,6,7]. For instance, P2P based solutions assume a relatively large number of clients, enabling the information to be present simultaneously at several locations rather than just on the original server. Therefore, usually the information will contain 'public properties' such as a TV broadcast.

Private, non-commercial video transmission has a couple of limitations which differ from public broadcasting – privacy and a small number of designated consumers of the information. The private properties of the information might turn irrelevant certain solutions, in which the user doesn't control the information flow (this might be the case for P2P or multicast) or unable to encrypt the information (which is unpractical for the private user in the case of live video).

Alternatively, the small amount of the specific information consumers makes it almost irrelevant to establish designated P2P networks to improve the transmission's quality.

Our work examines a novel approach toward resolving the issues mentioned above, by using a *small* number of coordinated agents (not exceeding 5) for uploading and/or downloading the information as plotted in Figure 1. This is based on the assumption that the domestic user has numerous ways of connecting to the internet (ADSL, WiFi, cellular, etc.) enabling the transmission of information or parts of it through different devices or connections.

 ISBN: 978-1-61208-166-3

The agents are assumed to be located geographically close to each other and therefore there might be a highly statistical dependency between the different links. For example, several participants in a family event will use their cellular phones to broadcast the live video to family members not attending the occasion. In such a scenario, some of the cellular devices will probably connect to the internet using the same base station and thus, sharing the cell resources. In addition, they are all affected simultaneously by the same radio interferences.

Still, despite the high statistical dependency, the multiple transmissions have the potential of using de-facto larger bandwidth than that available to a single transmission. Alternatively, it can supply various new options for dealing with delays and/or disruptions in one of the connections, and thus achieve continuous uniform quality of the received transmission. Our initial performance evaluation results indicate that the packet jitter can be easily improved with a competitive overhead factor relative to other methods.

The complexity of the method arises from the need for coordination of the agents, synchronization of the redundant information and choice of the most appropriate packets to assemble the received video. We show that this can be resolved with a simple algorithm and without a significant increase of information at the different connections. We implement agents' control mechanisms between the Web receiving/transmitting server and the agents as a function of current conditions of transmission. Thus, network or server resources consumption is minimized as the received quality becomes sufficient.

II. Related Work

Until recently most of the solutions for transmitting and watching video content over the internet were based upon on-demand services (such as YouTube). Over the last few years, research regarding Peer-to-Peer options were conducted, both as a solution for on-demand services, as well as for live video broadcasting. For instance, in [1] several typical P2P topologies are reviewed. P2P has mainly been used to minimize the number of connections (and transmissions) a server has to maintain simultaneously.

A key issue in P2P research is a fair distribution of resources among the network members and a minimization of the load from the original server. Solutions based on cooperative patches are presented in [2] and in [3] in order to handle re-transmit requests by other clients keeping different patches of information. Other aspects of P2P research deal with decreasing delay times as they are affected by the bandwidth available to the P2P network members [4].

A different approach, which is relevant to the commercial sector, is to optimize the various methods of transmission by dynamically 'activating' solutions. For example, CPM [5] is a solution which dynamically changes the transmission method of VOD as a function of video popularity, number of requests, numbers of clients, etc.

Our work resembles the concept of dividing a single broadcast into several transmissions and 'reunites' it back at the client side. This approach appeared as a possible solution toward some of P2P issues. For example, SplitStream [6] is an algorithm which intelligently builds P2P forests with the assumption that the application is responsible for splitting the transmission. Another relevant concept is to use multiple transmissions for encoding video signals in order to improve resolution and quality [7].

So far, a solution which assumes a relatively small number of statistically dependent agents collaborating in the transmission has not been suggested and examined. In this paper we suggest a new algorithm for controlling a small number of agents to provide better live video streaming quality.

III. Data Collection

As mentioned above, the complexity of the method arises from the need for coordination of the agents, synchronization of the redundant information and choice of the most appropriate packets to assemble the received video. Another issue concerns the fact that theoretical models assume no dependencies between the transmitting agents. However, this is not the case with 'real life' domestic clients where the agents are statistically dependent due to the close distance between them. Therefore, our method for evaluating the suggested solution is by measurements of real traffic, rather than theoretical analysis.

First, we transmit video using several agents in various conditions in order to collect data that will be used for evaluating different algorithms for splitting and joining the transmission. The transmission of the agents was done using LU60 of LiveU [8], using one to five cellular modems connected to three different cellular networks. Each agent has a different connection to the internet. Next, a feasible solution for splitting and joining is implemented. Finally, we evaluate the method potential performance using the data collected at the beginning. By that we are evaluating the potential for improving home video transmission for the domestic user. We record the received data with LiveU's server (LU1000) and also using 'Wireshark' software. We collect data which is relevant to parameters such as delay, jitter and retransmission ratio. Therefore, for each packet in each transmission from each agent we record the Packet Sequence Number and Time of Arrival (to server).

IV. Implementation

In this section we described our novel algorithm for controlling the agents. The server and the agents operate in a master-slave mode where the server is the master and the agents are the slaves. Regarding the algorithm for the coordinated transmission, we use a simple selection of the **best k** agents based on history of transmission of a segment consisting of N packets (N is equal to ten in our measurements). As long as the transmission requirements are satisfied, the selected k agents continue to transmit the next segment. If the requirements are not satisfied then a new competition is generated. In a competition, a new set of "best k" agents is selected based on transmission performance of a new segment.

The value of k is selected as the minimum value which satisfies pre defined performance transmission parameters. It varies between 1 and the maximum allowed number of

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-166-3

agents for the same session. Here we assume that the maximum allowed number of agents is five, so, k is in {1,…,5}. By on-line choosing a minimum value for k we reduce the transmission overhead; by limiting its value we actually limit the method overhead.

To start a new session the first agent sends a transmission request to the server. The request includes authentication information such as user identification and password. In addition, the request includes information on the requested performance parameters (e. g., Jitter), the number of expected cooperative transmitting agents for this session and the number of expected receiving (target) nodes. Once the authentication is completed, a new session is established with the first agent. Then, other agents can join the session.

An on-line analysis is performed to verify that the requirements are satisfied. To avoid waiting, it is based on the previous transmitted segment of N packets and not on the segment which is transmitted currently. The algorithm can be adjusted to longer server response time by shifting the analysis to even earlier arrived segments. However, a long gap between the current analyzed segment and the current transmitted segment result a long period of transmission with un-optimized set of k agents.

The server generates the joined video stream based on the arrived segments. For each packet, its first instance (with minimum arrival time) is placed in the joined file. This video is transmitted to any target node which is register to the specific session.

Similar to the server operation, using the same algorithms, a target node can activate a few receiving / transmitting agents to create a better video stream. Note that the master-slave operation can be done directly between the target node and the transmitting agents, so the web server is not necessary in this scheme and it can operate in a pure peer-to-peer manner.

The pseudo code of the algorithm is described below. For the simplicity of the presentation we assume that the live video transmission is longer than 2 segments and 5 agents have contact the server with request to join this specific session. EOF (end-of-file) is a flag set by a special message from the agents indicating that the next two segments are the last segments. We assume that all agents' registration is completed at the beginning of the video transmission. Obviously, this simple algorithm can be easily adjusted to the case where new agents perform registration after the beginning of the video transmission.

```
Void ServerMain()
Begin
   1: integer Seg_ind = 0;
   2: Agt_List new_list(); Best_Agt();
   3: Initiate(new_list);
   4: new_list.send(Seg_ind);
   5: Seg_ind++;
   6: new_list.send(Seg_ind);
   7: Seg_ind++;
   8: Best_Agt= Compete(new_list, Seg_ind-1);
   9: While ((Not EOF) &&
    (Transmission_quality(Best_Agt,
        Best_Agt.long(), Seg_ind-1))
   /* The "best" is still the best */
     Do{
   9.1:    Best_Agt.send(Seg_ind);
   9.2:     Seg_ind++;
         }
   10: If (Not EOF) /* The "best" is NOT good */
   10.1:      GOTO 4;
       Else { /* send last 2 segment and close */
   10.2:    Best_Agt.send(Seg_ind);
   10.3:     Seg_ind++;
   10.4:    Best_Agt.send(Seg_ind);
   10.5:   new_list.close();
          }
   11: return();
End.
```

The verification of the "Transmission quality" condition is done by verifying the pre defined ratio of packet retransmission threshold and pre defined packet jitter in the segment.

```
Bool Transmission_quality(B_Agt,k,prev_seg_ind)
begin
   0: quality = false;
   1: For (i=1 to N) do {
   1.1: Best[i]=
       Min(B_Agt[1][prev_seg_ind].pkt_arr_t(i), …,
       B_Agt[k][prev_seg_ind].pkt_arr_t(i));
   1.2: ReTrns[i] =
        Min(B_Agt[1][prev_seg_ind].ReTrns(i), …,
        B_Agt[k][prev_seg_ind].ReTrns
        (i));
      }
   2: For (i=1 to N-1) do {
   2.1:     If (Best[i+1]- Best[i]> Jitter) Then
               return(quality);
   2.2:     Sum_ReTrns += ReTrns[i];
      }
   3: Sum_ReTrns += ReTrns[N];
   4: If (Sum_ReTrns > ReTrns_TH) Then
         return(quality);
   5: quality = True;
   6: return(quality);
End.
```

Before a competition is performed, all registered agents are instructed to transmit two segments. The best k agents are selected according to the performance of the arrived first segment (again, packet loss ratio and jitter). If more than k agents fulfill the quality condition then the *first* set of such agents is selected as the "best". If none of the sets of k agents fulfill the quality condition then k is incremented. The selected agents are instructed to continue transmission. The other agents are instructed not to transmit.

```
Agt_List Compete(A_List, S_ind)
begin
   0: quality = false;
   1: k=1;
   2: Best_Agt = next set of k agents from A_list;
   3: quality = Transmission_quality(Best_Agt,k,
      S_ind);
   4: If (quality == false and more sets exist)
        GOTO 2;
   4.1: Else If (quality == false and no more sets
        of size k exist){
               k=k+1;
               if (k<6) GOTO 2;
               Else return ('error');
               }
   4.2  Else {return (Best_Agt)};
   End.
```

V. PERFORMANCE EVALUATION

The performance evaluation of this new method is not completed yet. However, we are able to provide some initial promising results.

In the performance evaluation of our novel method we consider the following additional competing methods: single transmission (that is, the way live video is transmitted today by the domestic user) and simple (not controlled or coordinated) multiple transmission of 2-5 agents. In simple multiple transmissions, for every packet sequence number, the server considers the first arrived instance. That is, the resulting arrival times are the minimum arrivals times of every packet. The analysis of the best k method was performed twice, once with jitter requirement of 13 msec. and once with jitter requirement of 25 msec.

Table I presents the Jitter statistics of the competing methods. Each line describes the average number of times that the arrival processes violate the corresponding jitter condition. Each complete video transmission consists of 50,000 packets. For example, in line number three, the jitter condition is “smaller than 13”, and the process “best k with parameter 25” violates this condition 1862 times in average out of 49,999 times (3.7%) while the process that use simple multiple transmissions of three agents violates this condition 3709 times in average out of 49,999 times (7.4%). As can be seen from this table, starting from jitter condition “smaller than 13” both best k processes outperform the other processes with significant small number of condition violation.

Regarding the average overhead, processes with one agent naturally have no overhead (factor 1), processes with two agents have overhead of factor 2 (every packet is transmitted twice), and so on. The best k process with parameter 13 has overhead factor of 2.7 and the best k process with parameter 25 has overhead factor of 1.66. The differences in the overhead factors are due to the fact that fewer competitions are generated when the best k algorithm requirement from the jitter is less demanding. In a competition all the potential 5 agents transmit a segment, thus, the overhead increases with the number of competitions.

TABLE I. JITTER STATISTICS

Jitter cond.	best k (25)	best k (13)	1 agt.	2 agt.	3 agt.	4 agt.	5 agt.
10	7254	7360	13213	9909	7657	5648	3905
13	1862	1402	7643	4984	3709	2842	2354
16	1767	1293	6877	4485	3291	2502	2027
19	1279	1006	5282	3503	2530	1894	1493
22	552	612	2891	2000	1498	1178	1017
25	517	582	2533	1825	1368	1086	938
30	444	537	2075	1536	1180	974	882
35	418	518	1891	1420	1094	911	835
40	408	508	1796	1353	1045	879	819
50	230	303	1007	842	756	743	785

VI. CONCLUSIONS AND FUTURE WORKS

This paper describes a new method to improve the quality of live video streaming for the private domestic sector. This method use a few agents installed for example, in the user laptop, smart phone and PDA. Upon registration, a server activates and coordinates the multiple transmission of the content from these agents in a way that improve the quality but with minimum overhead. In the downlink direction, the server can transmit the same stream to several agents. These streams can be joined again at the user computer using the same algorithm. We believe that this new web service is interesting as a complementary to sites such as MySpace and YouTube.

Future work includes:

- Additional analysis of the performance evaluation of this method.
- Improving the selection of the k transmitting agents. In our simple algorithm the *first* set of agents that fulfill the conditions is selected. We believe that different selection method can perform better.
- Dynamic estimation of actual performance conditions.

ACKNOWLEDGMENT

We thank LiveU, in particular, Noam Amram, for useful discussions, for allowing us to use the LU60 and for funding the required measurements over the cellular networks.

REFERENCES

[1] Liu, Y., Guo, Y., and Liang, C., A survey on peer-to-peer video streaming systems. In Peer-to-Peer Networking and Applications, 2008, vol. 1, no. 1, pp. 18–28.

[2] Guo, M., Ammar, M. H., and Zegura, E. W., 2004. Cooperative patching: A client based P2P architecture for supporting continuous live video streaming. In Proceedings of the 13th IEEE ICCCN. Chicago, USA, Oct. 2004, pp. 481-486.

[3] Guo, M. and Ammar, M. H., 2004. Scalable live video streaming to cooperative clients using time shifting and video patching. In Proceedings of the IEEE INFOCOM, 2004, pp. 1501–1511.

[4] Liu, Y., 2007. On the minimum delay peer-to-peer video streaming: how realtime can it be? In Proceedings of the 15th international conference on Multimedia, 2007, pp. 127-136.

[5] Gopalakrishnan, V., Bhattacharjeey, B., Ramakrishnan, K., Jana, R., and Srivastava, D., CPM: Adaptive video-on-demand with cooperative peer assists and multicast. In Proceedings of IEEE INFOCOM, Rio de Janeiro, Brazil, April 2009, pp. 91-99.

[6] Castro, M., Druschel, P. , Kermarrec A. M., Nandi A., Rowstron A., and Singh A., 2003. SplitStream: high-bandwidth multicast in cooperative environments. In Proceedings of the nineteenth ACM symposium on Operating systems principles, 2003, pp. 298-313.

[7] Schwarz, H., Marpe, D., and Wiegand, T., 2007. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. In IEEE Transactions on Circuits and Systems for Video Technology. September 2007, vol. 17, pp. 1103-1120.

[8] LiveU web site: http://www.liveu.tv/ accessed: June 2011.

 ISBN: 978-1-61208-166-3