Multilevel Client/Server Peer-to-Peer Video Broadcasting System

Dr. Subhi R. M. Zeebaree, Ibrahim M. I. Zebari

Abstract- Because of the high demands of video broadcasting systems and their applications that support different fields, these systems have to be modified to provide as maximum as possible services with high flexibility. The improvement can be relied on motion of the used cameras, increasing number of used hosts or using a tree of hosts. This paper presents the design and implementation of a proposed video broadcasting system. The proposed system makes to overcome the drawbacks of previous related systems to produce a system that includes: multi-sources (IP-cameras) via main-server to destinations/sources Clients/Services (CSs) which work as (middle-level), to clients (as final destination-level). This system built depending on client/server principles with peer-to-peer technology that provides bottleneck avoidance. The video signals captured by IP-cameras send to the main-server to be displayed and broadcasted to level-one (middle-level). These signals displayed again and rebroadcasted to level-two (final clients) and displayed there. The proposed system is friendly used with as maximum as possible of flexibility and full optional controlling to get complete features of the video broadcasting systems. Hence, this system enables the administrator to monitor the dataflow from main-server to CSs then to clients during the broadcasting. Also, the broadcasted video signals can be recorded, compressed and embedded with demand texts during the broadcasting process. The results are tested depending on different proposed scenarios, also tested using different distances between the source and destination parties. Adding to that, the obtained results are compared with Wireshark tools as additional evaluation test. The algorithms of this software application are designed and implemented by Visual C#.Net language with using Microsoft Framework.Net.

Index Terms— Video broadcasting, Streaming, Video Compression, Audio Compression, IP-camera, Client/Server, peer-to-peer, LAN, Clustering.

1 INTRODUCTION

Video has a big role for communications since several decades. Data transmissions that represent multimedia applications from point-to-point can be used for video conferencing or other visual telephone with secure and enhance style [1]. One of the famous approaches for exchanging information and knowledge is video sharing. In order to reduce the redundant data produced from cameras, it is preferred to apply video compression techniques [2]. Video transport based on Internet Protocol (IP) technologies has been depended in wide range over IP-networks with high bandwidth [3].

In 2008, Hyen K. [4], described the design and implementation of video conference system over peer-to-peer networks. Server traffic was controlled in this system. In 2010, Dakshayini M. and Gopala K. [5], proposed an efficient client-to-client system for video streaming among the clients. The system is a unicast communication depending on chaining technique. In 2012, Zhang S. et al. [6], proposed a distributed architecture for high definition multi-view video surveillance system. It adopts a modular design where multiple intelligent IP-based video surveillance cameras were connected to a local video server. In 2013 H. Rajab [7], presented a system for broadcasting video-signals from multi-sources to multi-destinations using fundamentals of clients/server. The system made the improved application software to be used friendly with flexibility and optional controlling.

As a conclude comparison between our proposed system and the previous works, in this paper, the proposed system provides all facilities stated in the previous works that can be considered as an aggregate system of all the previous works. Hence, this system has multi-source, multi-level of destinations (level-1 receives signals from server of multi-source and level-2 receives signals from level-1), online receiving the incoming signals from the sources, online broadcasting them to level-1 of multi-destinations and online rebroadcasting them to level-2 of multi-destinations Clients. Also, many other flexible options provided with this system to be more efficient and powerful.

2 BACKGROUND THEORY

The possible client/server topological designs and strategies are: single-client/single-server that considered as one client directly connected to one server. Multiple-clients/single-server that considered as several clients directly connected to only one server. Multiple-clients/multiple-servers that considered as several clients connected to several servers [8].

Multimedia real-time streaming can be done through 3 common techniques: first one is unicast stream between one-server and client (or clients). Via this technique, each client receives a distinct stream and only those clients that request the stream receive it [9]. Second technique is multicast that links single/multi servers with multi-clients. This is mean that single video-signal, is delivered simultaneously to multiple users. Through the use of special protocols, the network is directed to make copies of
the video stream for every recipient [10]. Third technique is
broadcast that depends on linking single server with many
(all) clients. This technique is used for transmitting data to a
large number of host systems simultaneously. Broadcast
networks are multi-access networks with broadcast
abilities. LANs such as Ethernet or Token Ring are

The depended approach in this paper collects the
abilities of all above three techniques. So, the proposed
video broadcasting system assembles the functionality of
unicast, multicast and broadcast techniques. Hence, video
streaming is used as pointing to all of these techniques.

Streaming video is a process for sending video and
audio content to a user over an IP network. The sent signal
watched immediately by end-users, just like watching a TV
network broadcast [12]. Two categories of video streaming
can be depended, based on the broadcasting technology
used (Centralized and non-centralized) as shown in Fig. 1,
[13]:

Centralized: the server is responsible for accessing, storing
and transferring data. While the client, is responsible for
decoding and playing the received video-signal.

Non-centralized: the clients actively participate in sending
videos to other clients. In this category solutions based on
peer-to-peer networks are surveyed which enable peers to
collaborate without the need of a centralized component.

In this paper, the proposed system merges both of above
categories in one broadcasting technology to have the
advantages of the both.

3 ORGANIZATION OF THE GENERAL SYSTEM DESIGN
The proposed system is organized with four hardware-sides
(IP-cameras, main-server, CSs and Clients).

Server-side consists of one host (main-server) while clients
are grouped in two levels. First level represents CSs
consists of Y-hosts (for the test Y is considered equal to 3).
These hosts work as clients with respect to the main-server
and as servers with respect to the fourth-side (i.e. second
level). Second level represents clients which consist of Z-
hosts (for the test Z equals to 6). Fig. 2, represents the
general organization of the proposed system.

3.1 Mechanism of the Proposed System Operation
Main-server software usually requires special system
privileges. Because it has many responsibilities which are:
receiving video signals from all IP-cameras. Then
displaying them on its viewers (viewer is a window screen
that is used by main-server to display video signal on it),
this system main-server used three viewers. Adding to that,
main-server has many special control-options: First control
to manage video signals for all viewers in proposed system
such as; turn on/off video signals, set mute/unmute sound,
recording and stopping video signals. Second control-
options belong broadcasting video signals such as;
broadcasting and sending permissions. Third control-
options related with overlay-text on video signals. Fourth
control-options concerned with moving IP-cameras to all
directions.

The other part of proposed system is IP-camera part.
The IP-cameras used in this system have two image
resolution-modes; one is the video graphics array VGA
(640*480 pixel) and the other is Quarter video graphics
array QVGA (320*240 pixel). The default resolution is
(640*480 pixel).

The GUI of main-server illustrate in Fig. 3- a. While
software of CSs at level one has dual mode operations.
During the first mode CSs working as clients; by receiving
video signals from main-server and displaying them on its
viewers. While during second mode CSs work as servers;
by rebroadcasting the received video signals to other clients
in level two. The CS software at level one has limited
control-options; broadcast/non-broadcast video signals to
other clients at level two, Fig. 3-b represents GUI of CS. In other side the responsibility of client software at level two is receiving video signals, and then displaying them on its viewers. The GUI of client at level two illustrate in Fig. 3-c.

### 3.2 Stages of the Proposed System

There many stages depended to achieve the proposed system. Hence, getting a successful system that provides the sending/receiving video signals through one or more levels.

The stages are started at Preparing Main-Server Host (PMSH). Finding Connected IP-Cameras (FCIPCs) at main-server host. Preparing CS Hosts of level one (PCSHs). Finding Connected Client Hosts (FCCSHs) at Cs. Start Broadcasting Video Signals (SBVSs) at main-server. Applying Special Control Options (SCOs) or Common Control Options (CCOs). Start Broadcasting Video Signals (SBVS) at Cs. Applying Common Control Option (CCO) at Cs. Last stage is the Monitoring Interface Connection Maps (MICM) that displays all connections of the system including all hosts and cameras. Fig. 4, shows the main stages of proposed system.

### 4 IMPLEMENTATION RESULTS OF THE PROPOSED SYSTEM (EVALUATION CASES)

After completing system designing with its two parts (HW and SW), it is necessary to be evaluated with suitable evaluation-cases. These cases start from receiving video signals of IP-cameras at main-server and broadcasting them to Cs at level one, and then Cs do the role of main-server to rebroadcast them to the clients at level two.

1) Stage one: Main-server receiving video signals from IP-cameras: there are two main scenarios for receiving video signals from IP-cameras.
   a) One to One receiving (OO): receiving signal of only one IP-camera at main-server.
   b) Many to One receiving (MO): receiving signals of more than one IP-camera at main-server.

2) Stage two: Main-server broadcasting video signals to Cs: there are four scenarios in this case:
   a) One to One to One broadcasting (OOO): only signal of one IP-camera is broadcasted from main-server to one CS at level one.
   b) One to One to Many broadcasting (OOM): only signal of one IP-camera is broadcasted from main-server to more than one CS at level one.
   c) Many to One to One broadcasting (MOO): signals of more than one IP-camera are broadcasted from main-server to one CS at level one.
   d) Many to One to Many broadcasting (OOM): signals of more than one IP-camera are broadcasted from main-server to more than one CS at level one.

3) Stage three: Cs at level one receiving and rebroadcasting video signals to Clients: eight scenarios are related with rebroadcasting signals received from main-server by Cs:
   a) One to One to One rebroadcasting (OOOO): only signal of one IP-camera is rebroadcasted from many Cs at level one to one related client at level two.
   b) One to One to Many rebroadcasting (OOOM): only signal of one IP-camera is rebroadcasted from many Cs at level one to more than one related client at level two.
   c) One to One to Many rebroadcasting (OOMO): only signal of one IP-camera is rebroadcasted from many Cs to more than one related client.
   d) One to Many to One rebroadcasting (MOOO): signals of more than one IP-camera are rebroadcasted from one CS at level one to one related client at level two.
   e) Many to One to One rebroadcasting (MOOO): signals of more than one IP-camera are rebroadcasted from many Cs at to more than one related client.
   f) Many to One to Many rebroadcasting (MOMM): signals of more than one IP-camera are rebroadcasted from one CS at level one to more than one related client.
   g) Many to Many to One rebroadcasting (MOMO): signals of more than one IP-camera are rebroadcasted from more than one CS to one related client.
   h) Many to Many to Many rebroadcasting (MOMM): signals of more than one IP-camera are rebroadcasted from more than one CS at level one to more than one related client at level two.

### 4.1 Results Obtained by Applying some Scenarios Addressed in This Paper

Figs. 5 to 12, represent amount of dataflow for some of the scenarios depended in this paper as evaluation case.
Fig. 5. Dataflow of main-server using (OO and MO) scenarios.

Fig. 6. Dataflow of main-server using (OOO and OOM) scenarios.

Fig. 7. Dataflow of main-server using (MOO and MOM) scenarios with 3-IP-cameras.

Fig. 8. Dataflow of 1-CS using (OOOO and OOOOM) scenarios with 1-IP-camera.

Fig. 9. Dataflow of 3-CS using (OOOO and OOOOM) scenarios with 1-IP-camera.

Fig. 10. Dataflow of 1-CS using (MOOO and MOOM) scenarios with 3-IP-cameras.

Fig. 11. Dataflow of 3-CS using (MOOO and MOOM) scenarios with 2-IP-cameras.

Fig. 12. Dataflow of 3-CS using (MOOO and MOOM) scenarios with 3-IP-cameras.
4.2 Results of Receiving video signal from different distances test

In order to evaluate the efficiency of the proposed system for sending and receiving video signals from a distance far of the camera’s position, this system is tested from different distances. Table 1, illustrates receiving video signal from one IP-camera at main-server from two different distances test. It is clear that there is a little difference between results of the two distances.

<table>
<thead>
<tr>
<th>Time in Sec</th>
<th>Received Data (KB)</th>
<th>2 Meters</th>
<th>100 Meters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>314</td>
<td>314</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>629</td>
<td>628</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>943</td>
<td>942</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1233</td>
<td>1231</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1535</td>
<td>1534</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1849</td>
<td>1848</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2171</td>
<td>2169</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2478</td>
<td>2476</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2792</td>
<td>2791</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3107</td>
<td>3105</td>
<td></td>
</tr>
</tbody>
</table>

4.3 Results of some compression techniques tested by this System

There are many video codec algorithms can be used for the compression. Some of them are selected for this system and applied on video signals when recording and broadcasting video signals. This system has the capability of using the installed audio/video compression techniques for compression purposes. This feature gives more power to this system which produce broadcasted video signals with as minimum as possible of frames-redundancy. Table 2, represents the obtained results using several codec algorithms used when video signals are recorded. Figs. 13 and 14, represent the dataflow of different compression techniques that illustrated in Table 2.

<table>
<thead>
<tr>
<th>Time in sec</th>
<th>Type of Compression Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AVI (Compressed) ffdshow video encoder MPEG Compress DivX 6.9.2 Codec Microsoft MPEG-4 Video Codec V2</td>
</tr>
<tr>
<td>1</td>
<td>2815 326 91.45 136 209 94.76 210 94.59</td>
</tr>
<tr>
<td>2</td>
<td>7640 654 91.40 161 209 94.77 210 94.53</td>
</tr>
<tr>
<td>3</td>
<td>11104 981 91.40 154 209 94.77 210 94.53</td>
</tr>
<tr>
<td>4</td>
<td>13199 326 91.37 131 209 94.76 210 94.57</td>
</tr>
<tr>
<td>5</td>
<td>10993 1638 91.38 163 209 94.74 210 93.59</td>
</tr>
<tr>
<td>6</td>
<td>22791 1091 91.53 167 209 94.71 210 94.52</td>
</tr>
<tr>
<td>7</td>
<td>26386 2294 91.76 225 209 94.74 210 94.50</td>
</tr>
<tr>
<td>8</td>
<td>30380 2444 91.82 262 209 94.73 210 94.49</td>
</tr>
<tr>
<td>9</td>
<td>34175 2810 91.78 297 209 94.70 210 94.49</td>
</tr>
<tr>
<td>10</td>
<td>37973 3215 91.74 328 209 94.70 210 94.49</td>
</tr>
</tbody>
</table>

4.4 Evaluation Results of the Proposed System Using Wireshark Tool

The results of proposed system evaluated by Wireshark when receiving data from IP-cameras and broadcasted to CSs at level one. Taking in consideration that Wireshark tool is a network packet analyzer and it is used to capture the packets during live migration. The captured packets are stored in the Wireshark tool and can be used for the analysis. It runs on various operating systems such as Microsoft Windows and Unix. Tables 3 and 4, illustrate comparisons between results of the proposed system and those of Wireshark tool.

<table>
<thead>
<tr>
<th>Application type</th>
<th>Main-server received data (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>This Paper</td>
<td>One IP-camera Two IP-cameras Three IP-cameras</td>
</tr>
<tr>
<td>Wireshark Tool</td>
<td>4147 8246 12142</td>
</tr>
</tbody>
</table>

Fig. 13. File-size of recorded one IP-camera-signal using AVI, ffdshow video encoder and DivX 6.9.2 codec compression-techniques with default resolution (640*480 pixel) through 10 seconds

Fig. 14. File-size of recorded one IP-camera-signal using AVI, M-JPEG compressor, Micro Windows mpeg-4 video codec compression-techniques with default resolution (640*480 pixel) through 10 seconds

Table 1

Receiving video signal from one IP-cameras from two different distances test.

Table 2

Comparison among several codec algorithms and AVI

Table 3

Comparison between results of proposed system and Wireshark when receiving data for 10 seconds.


**5 CONCLUSIONS**

The most important points concluded from this work can be summarized as bellow:

1. An efficient system is proposed for video broadcasting signals from Multi-Sources to Multi-Destinations with multi-levels. This system was designed via LAN network depending on the principles of client/server mixing with principles of peer-to-peer technology.

2. The software application of this system is built with full flexibility and friendly used with the capabilities of recording, compressing and text addition to the broadcasted signals through or without broadcasting process. Also (control and data)-massages are depended.

3. Using IP-cameras instead of USB-cameras provided the abilities of moving IP-cameras direction during video capturing (horizontally and vertically). Also, provides the ability of video capturing from far distances.

4. Because of the occurrence of bottleneck with increasing number of clients, the principles of multi-level depended in this system by inserting a middle-level of CSs.

5. From the obtained results, it is clear that the rate of dataflow increased with increasing number of sources (IP-cameras) or destinations (CSs or Clients).

**6 REFERENCES**


