Scalable Distributed Standard Definition Video Conferencing System: Architecture and Forwarding Model

Suiming Guo
CERNET Research Center,
Tsinghua University
Beijing, 100084, P. R. China
guosm11@mails.tsinghua.edu.cn

Congxiao Bao
CERNET Research Center,
Tsinghua University
Beijing, 100084, P. R. China

Xing Li
CERNET Research Center,
Tsinghua University
Beijing, 100084, P. R. China

***Abstract*—Videoconferencing system is widely and successfully used on the Internet and standard/high definition videoconferencing is a new heated topic in research. However, researches in various videoconferencing systems (e.g. Access Grid, DVTS, VLC, and Ultra Grid) are faced with the problem of scalability. Bandwidth is the primary factor that limits the system capacity. In this paper, we propose a scalable and distributed standard definition videoconferencing system and design its architecture as well as forwarding model. We use several data stream transmitters to cooperatively forward video streams and design the system architecture, components' behaviors, and forwarding models on both control and data plane, with reasonable load allocation and balance. We also implement our idea based on DVTS as a multi-point standard-definition distributed DVTS Plus and test it on CERNET. Experiments on CERNET show that distributed DVTS Plus now supports higher system capacity, easier deployment and better scalability.**

Keywords-videoconferencing; distributed system; load balancing; scalability; large-scale conference management; DVTS

I. INTRODUCTION

Researchers around the world have done intensive researches about videoconferencing systems. Traditional videoconferencing systems support CIF video formats (with definition of 352*288), such as H. 323 and Access Grid. With the great increase of network bandwidth and computer processing ability, standard/high definition video transmission becomes another heated topic. Representative systems of standard/high definition videoconferencing include DVTS by WIDE Project in Japan (bandwidth cost 30Mbps), HDTV by GIST and KAIST in Korea (30Mbps), MPEG-2 based VLC Media Player by VideoLAN (22Mbps) and UltraGrid (1.4Gbps) by ISI of University of Southern California. All these video transmission tools realized point-to-point standard/high definition video transmission, but still were not able to host large-scale standard/high definition videoconferences and faced problems of real time, terminal scalability and network scalability.

CERNET Research Center of Tsinghua University proposed a DV-based extensible video transport terminal based on DVTS and offered multi-point standard definition videoconferencing by combining DV and MJPEG video streams. In this design, DV stream is used to transmit standard-definition video of certain one conference attendee, and MJPEG video streams are used to transmit low-definition videos of all other attendees. In this way conference attendees could have not only a clear, detailed representation of certain one of them, but also an overall representation of the whole conference. The combination of MJPEG and DV streams could satisfy users' various requirements, while at the same time doesn't bring huge pressure on the network and computer processor. Based on this idea, a multi-point standard-definition videoconferencing system could be implemented.

Regardless of which video transmission scheme we choose, the bandwidth cost of videoconferencing (varying from 30Mbps to several Gbps) is the primary problem to consider. The capacity of videoconferencing system, i.e. how many conference attendees it could accommodate simultaneously, is bandwidth-limited, and this seriously limits its application. A video stream transmitter with 1Gbps network adapter could only support 500/X attendees, if each attendee needs X Mbps bandwidth to send or receive their videos.

Scalability is a related problem. Videoconferencing system with single video stream transmitter would meet problems when we extend it to accommodate large-scale conference because of limited capacity. We need a new design to improve system scalability and broaden its applicable scenarios.

We also need to take into consideration load balancing. Delay, bandwidth and connection rate differ considerably when users from different areas and countries try to connect to a single transmitter to transmit video streams. In the meantime, this difference significantly lowers user experience. Our tests show that video delay differs from tens to hundreds of milliseconds. In times of bad network condition, video jitter occurs when considering users from different areas and countries. Therefore, we need to introduce multiple transmitters to serve users from different areas and to guarantee the utilization and performance (i.e. delay, jitter and bandwidth) of each transmitter by load balancing.

In this paper, instead of a centralized videoconferencing system, we propose a general design of the architecture and forwarding model of a distributed scalable standard-definition videoconferencing system. We then implement this DV-based,

multipoint, distributed video transmission DVTS Plus system and conduct extensive experiments on CERNET (China Education Research Network, the second largest research network in the world).

II. DESIGN OF DISTRIBUTED VIDEOCONFERENCING SYSTEM

A. *Architecture and Design Criteria*

Generally speaking, a typical centralized (with single video stream transmitter) videoconferencing system consists of three parts: clients (conference attendees use it to join the conference session), one conference center (responsible for signaling and controlling) and one video stream transmitter (responsible for forwarding client-client audio/video streams). Fig. 1 shows the architecture of a typical centralized system.

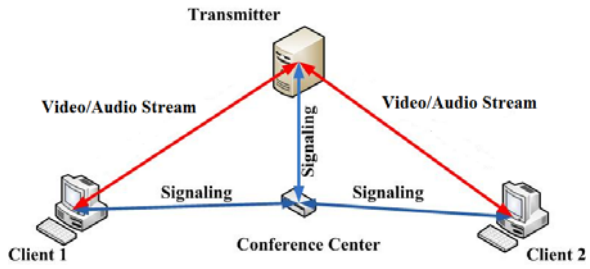

Figure 1. Architecture of a Centralized Videoconferencing System

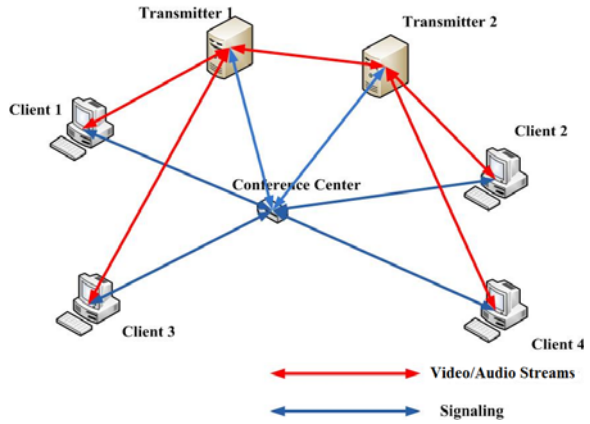

Figure 2. Architecture of a Distributed Videoconferencing System

As mentioned above, the number of transmitters limits the system capacity and scalability. In our proposed distributed videoconferencing system, we allow more than one video transmitters (for the consideration of scalability, we assume the number to be N), and they work collaboratively. Fig. 2 shows the architecture of the distributed videoconferencing system (as an example, we let N=2 and the number of clients is 4).

Below we explain the architecture of a distributed videoconferencing system.

- The system consists of three parts: one conference center (responsible for the collection of control information and signaling), N transmitters (responsible for transmitting audio/video streams; N≥1), and M clients (used by conference attendees to participate in conference; M≥1).
- All of the M clients register their own IP address and ports on conference center. The conference center then distributes registered information to N transmitters.
- Each client connects to one (and only one) of the N transmitters (e.g. client *i* connects to transmitter *j*). So client *i* sends its audio/video streams first to transmitter *j*; and transmitter *j* then forward audio/video streams to other transmitters or clients.
- The topology of these N transmitters is of full-mesh connectivity. This means that any one of the N transmitters could send video streams to any other one transmitter directly, without any relay or intermediate transmitters.

We also explain the design criteria of a distributed videoconferencing system.

- The same data could only have one copy to be sent between transmitters. Copy and distribution of audio/video streams only happen in the link between transmitter and clients.
- We must avoid loss and duplication of audio/video in forwarding, otherwise it would give rise to discontinuity or jitter.

B. *Load Balancing: Connection between Transmitters and Clients (IP-ASN-Country mapping)*

Client-client video streaming (data transmission) is carried out by the direct transmission on the only transmitter in a centralized system. In distributed videoconferencing system, however, we need to first establish a connection relation (i.e. which transmitter should be used by certain client to transmit its video stream) between transmitters and clients before starting transmission. For instance, in Fig. 2 we say that Client 1 and Client 3 are connected with Transmitter 1, i.e. video streams from either Client 1 or Client 3 would be only sent to Transmitter 1.

The determination of the connection relation between transmitters and clients is also a process of load distribution. N transmitters are responsible for sending/receiving the load generated by M clients. This load distribution leads to stationary load balancing in our distributed videoconferencing system, i.e. the performance of network connection (including delay, jitter, and bandwidth) between clients and corresponding transmitters could satisfy users' requirement, and the amount of video streams to be forwarded by any one of the N transmitters does not exceed its capacity.

The distributed system we propose in this paper is currently running in academic network such as CERNET (China Education and Research Network), where geographic location has great impact on the rate, bandwidth and delay of the network connection between two computers. Generally speaking, the performance of network connection could be guaranteed if the two computers belong to the same AS; if they don't, then the nearer the two computers are, the more guaranteed the network connection performance is. Thus, in our distributed system, we propose that the connection relation

should be determined by geographical proximity. To elaborate on it, we first use DNS query to get the corresponding AS Number (ASN) of the client's IP address [8], and also the corresponding country the client belongs to (e.g. the ASN of IP address 202.38.101.25 is 4538, and AS4538 belongs to China). We then choose the transmitter that shares the ASN with the client (or that is geographically nearest to the country of the client) to serve the client. In short, we use an IP-ASN-Country mapping to determine the connection between transmitters and clients.

Connection relation determined in this way is closely related to the conference attendees' topology, and in most cases the rate and bandwidth between client and its corresponding transmitter could be guaranteed. .

C. Forwarding Model

The only one transmitter in a centralized system has very simple behavior: it receives data flow (video stream) from clients, and then directly sends them out to destinations. By comparison, transmitters' behavior and forwarding model in a distributed system is much more complicated.

Consider the general situation where there are N transmitters and M clients (one conference center) in the system. We use an array t[M] to store the connection relation between clients and transmitters, in which $t[i]=S_j$ (i=1,2,…,M, j=1,2,…,N) means that clients i connects to transmitter S_j. Now we consider the process of sending and receiving video streams between client i_1 and i_2 (i_1, i_2=1,2,…,M). Let $D(i_1, i_2)$ denote the data path from i_1 to i_2 (the nodes the video streams from client i_1 to i_2 pass by). We obtain,

$$D(i_1,i_2)=\begin{cases} i_1 \to t(i_1) \to i_2, if\ t(i_1)=t(i_2) \\ i_1 \to t(i_1) \to t(i_2) \to i_2, if\ t(i_1)\neq t(i_2) \end{cases}$$

We must note the sequence of generating t[M] and $D(i_1, i_2)$. Clients logon to the video conference and register their own IP address and ports in the conference center. The conference center then use IP-ASN-Country mapping to determine the connection relation, thus generating t[M]. The data path $D(i_1, i_2)$, on the other hand, is determined real-time when sending/receiving video streams based on t[M]. In our distributed system, stationary, instead of dynamic load balancing is supported, so t[M] and $D(i_1, i_2)$ remain unchanged during a conference.

We now discuss the micro behavior of transmitter. The so-called "micro behavior" is contrary to the "macro behavior" mentioned above, and it denotes the procedure of determining the data transmission destination when video streams arrive on the transmitter. Assume each client sends K video streams, and we use "channel" to transmit each of them. One channel is responsible for the transmission of one video stream.. As there are M clients in the system, each transmitter needs to maintain KM channels. Let C_m denotes the mth channel (m=1,2,…,KM) on the transmitter, then the nth video streams (n=1,2,…,K) of the ith client uses the channel with m=(i-1)K+n. Note that EACH transmitter must maintain ALL KM channels globally, instead of just maintaining channels related to clients connected to the transmitter itself.

Transmitters follow certain rules in forwarding video streams. All transmitters' behaviors are based on channels, instead of the transmitter itself. In general, transmitter's behaviors on channels depend on the source of data flow: whether the video stream is directly from clients or indirectly from other transmitters' forwarding. We now give out transmitter's forwarding rules in details. Assume that the source of video stream is client i_1 and the destination is client i_2. The transmitter in question is S_j. We have the following rules.

$$\begin{cases} forward\ to\ i_2\ (t[i_1]\neq S_j, t[i_2]=S_j) \\ discard\ the\ packet (t[i_1]\neq S_j, t[i_2]\neq S_j) \\ forward\ to\ i_2 (t[i_1]=S_j, t[i_2]=S_j) \\ forward\ to\ t[i_2] (t[i_1]=S_j, t[i_2]\neq S_j) \end{cases}$$

We still use the relation in Fig. 2 as an example (N=2, M=4) to explain transmitters' behaviors in the distributed system. For the connection relation in Fig. 2, we obtain t[M] as below,

$$t[1]=S_1,\ t[2]=S_2,\ t[3]=S_1,\ t[4]=S_2.$$

Client 1 and 3 are connected with transmitter 1; client 2 and 4 are connected with transmitter 2. The corresponding data path $D(i_1, i_2)$ could be easily determined

As an example, we let K=2. Since M is 4, each transmitter maintains 8 channels. Channel 1 and 2 are responsible for transmitting client 1's video streams; client 3 and 4 are responsible for transmitting client 2's video streams; client 5 and 6 are responsible for transmitting client 3's video streams; and client 7 and 8 are responsible for transmitting client 4's video streams.

To illustrate transmitter's video streams forwarding rule, we use several cases to make it clearer: (1) channel 1 on transmitter 2 receives the 1st video stream from client 1, so that i_1=1. And if i_2=2, because $t[1]\neq S_2$ and $t[2]=S_2$, this video stream is then forwarded to client 2; (2) channel 1 on transmitter 1 receives the 1st video stream from client 1, so that i_1=1. And if i_2=3, because $t[1]=S_1$ and $t[3]=S_1$, this video stream is then forwarded to client 3; (3) channel 1 on transmitter 1 receives the 1st video stream from client 1, so that i_1=2. And if i_2=2, because $t[1]=S_1$ and $t[2]=S_2$, this video stream is then forwarded to transmitter 2.

III. Implementation of Distributed DVTS Plus and Experiments on CERNET

Based on the design of architecture and forwarding model above, we implemented a DV-based, multipoint and distributed standard-definition videoconferencing system DVTS Plus and conducted extensive experiment on CERNET (the largest education network in China). This system has also been successfully used in APAN medical group meetings and Internet 2 meetings. Fig. 3 shows how it works.

DVTS Plus combines DV and MJPEG video streams, using DV stream to transmit standard definition video (720*480) and MJPEG stream to transmit low definition video

(180*120). Each conference attendee sends and receives one DV stream and multiple MJPEG streams. As to the bandwidth cost, it takes 30Mbps to transmit DV stream (whereas the bandwidth cost of MJPEG streams is negligible). The capacity a single transmitter could offer is about 15 (i.e. 15*2*30=900Mbps≈1Gbps). In our distributed DVTS Plus, the system capacity is 15N when we use *N* transmitters.

Figure 3. How DVTS Plus works

We tested our distributed standard-definition DVTS Plus on CERNET, using N=2 and M=3 (i.e. using two transmitters, three clients and one conference center in the scenario). Fig. 4 gives the test topology.

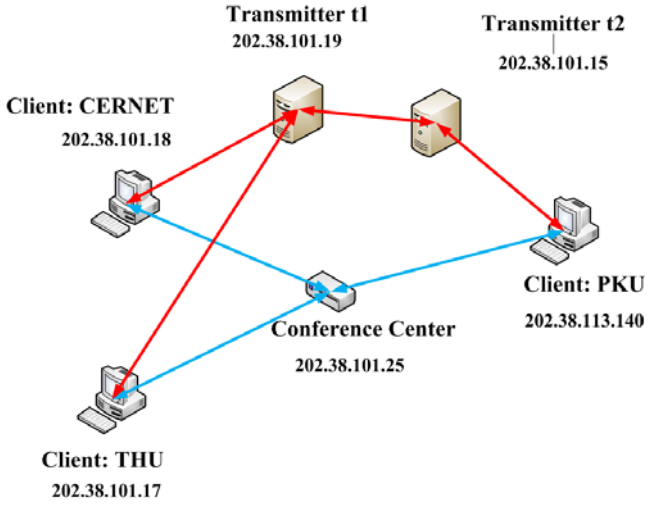

Figure 4. Topology of our Testing Environment

We also implemented a real-time network monitor based on header of RTP packet used to encapsulate DV and MJPEG streams and deploy it on transmitters. The monitor could log packet loss and duplication rate in real-time. In our test, both DV and MJPEG streams are very smooth, without any obvious time delay (i.e. smaller than 100ms) and jitter. The test has been conducted continuously for two weeks to test stability and we found no instability issue. The monitor gave the result that the packet loss rate is smaller than 0.0001% and duplication rate smaller than 0.00005% (see Fig. 5). All these results show that distributed DVTS Plus could run smoothly and function correctly in real network.

```
[root@localhost NetMonitor]# ./NetMonitor
Total Packet: 25378140
Loss Packet Count: 25
Duplicate Packet Count: 11
Packet Loss Rate: 0.000099%
Packet Duplicate Rate: 0.000043%
```

Figure 5. Real-time network monitor results

We also have actual conference that accommodated 20 attendees (i.e. 20 clients) using 2 transmitters. The conference was held successfully and the video was very smooth without observable time delay (i.e. time delay longer than 100ms) and jitter.

IV. Conclusion and Future Work

In this paper, we proposed a new distributed DVTS Plus based on DVTS and existing single-transmitter DVTS Plus made by CERNET Research Center in Tsinghua University. We designed the architecture and components' behavior of distributed DVTS Plus and implemented and tested the system on real network (CERNET). Experiments on CERNET show that our distributed DVTS Plus could not only increase system capacity, enhance scalability, but also keep system stable and keep packet loss as well as duplication rate on a satisfying level. This makes it possible to apply distributed DVTS Plus to various scenarios.

Our future work would still lies on the architecture improvement. We are going to introduce the concept of Named Data Networking to support name-based routing and improve scalability and routability. We would also introduce support for IPv6 and high definition videoconferencing.

Acknowledgment

The authors would like to thank DVTS Plus users who have continuously given us meaningful feedback to improve our videoconferencing tool.

References

[1] Access Grid Website [2012-07-17]. http://www.accessgrid.org

[2] Akimichi Ogawa, Katsushi Kobayashi, Kazunori Sugiura, Osamu Nakamura, Jun Murai, Keio University, Japan (2000). Design and Implementation of DV based Video over RTP. Packet Video 2000.

[3] DVTS - DV (Digital Video) over IP: http://www.sfc.wide.ad.jp/DVTS/

[4] VideoLAN – Free Software and Open Source Video Streaming Solution for every OS. [2012-07-17]. http://www.videolan.org/

[5] UltraGrid – A High Definition Collaboratory. [2012-07-17]. http://ultragrid.east.isi.edu

[6] Dongtao Liu, Congxiao Bao, Xing Li, Xuan Zhang. (2007). DV-based Extensible Video Transport Terminal for Standard Definition Video Conference. 2007 International Conference on Intelligent Pervasive Computing.

[7] IP to ASN Mapping: http://www.team-cymru.org/Services/ip-to-asn.html

[8] RTP: A Transport Protocol for Real-Time Applications. RFC 3550, July 2003. IETF.

[9] Dongtao Liu. Design and Implementation of a DV-based Extensible Standard Definition Videoconferencing System. Beijing: Tsinghua University, [Master Thesis] [in Chinese]