A Service Oriented Architecture for a Multi Channel Internet TV System

Md. Tanvir Al Amin, Imranul Hoque, Md. Yusuf Sarwar Uddin
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology,
Dhaka, Bangladesh.
tanviralamin@gmail.com, imranulhoque@cse.buet.ac.bd, yusufsarwar@cse.buet.ac.bd

Abstract

Transmitting TV programs over the Internet has always been a great issue in multimedia service. With the advent of broadband internet connectivity, real time video and audio delivery on the Internet is getting popular day by day. Although the industry is making great and highly publicized plans for future (inter)national video on demand, there is lack of robust, scalable and interoperable architecture for Internet Television System. In this paper we present a Service Oriented Architecture for a Multi Channel Internet Television System. TV Program Servers, which are capable of capturing and transmitting media, registers with the Broker Server. The Internet TV Clients query the Broker Server, which maintains a list of available program servers and a complex data structure about the clients already connected. The strength of this architecture is that, it can be implemented over the existing network and no special expensive hardware setup is required. The application of the service-oriented paradigm makes the registration, composition and discovery of services efficient, and hence the system is perfectly suitable for medium to large scale Internet Television.

Keywords: Internet TV Client, Internet TV System, TV Broker Server, Service Oriented Architecture, Service Tree, TV Program Server.

I. INTRODUCTION

Because of the increase in Internet connection speeds and the total number of people online, and the decrease in connection costs; it is increasingly common to find traditional television content, accessible legally over the Internet. Internet TV is a concept where people can watch TV programs of several channels transmitted to their PC over the internet. Where several proprietary technologies exist for Internet TV systems, no research standard is defined for the maintenance and service deployment model of such a system.

Transmission of TV programs via Internet is a low cost approach for delivering TV channels to clients. Obviously, this approach is far less expensive than typical air based TV channels. Specialized Internet-only TV channels are also possible. While a TV Program Server is capable of providing these services with available technologies [1], [3], [4], there are still some research issues concerning the delivery of TV Programs over internet in an optimal way.

Audio and video related applications over Internet have always been bandwidth dependent. As the number of clients increases, pressure on the underlying network increases. Consequently, it is necessary to use some optimization techniques to implement such system. An Internet TV system has its own sets of challenge issues [6] that motivate us to find a general and low cost way to provide service.

Moreover, it is necessary to define the service model and policy – which not only deals with the TV Programs, but also ensures network integrity, connection and availability to the clients, takes care of the Quality Control and efforts to minimize network pressure and congestion.

In this paper, we propose an architecture for an Internet TV System based on Web Services [2] with its design and implementation issues. This is a distributed system, and, anyone having client software and access to internet will be able to view live television programs of the available channels without any external setup for it.

The proposed system depends on the interaction of the working subsystems. The TV Program Servers (TPS) are capable of capturing and transmitting TV programs in real time as streaming audio-video. These will form cluster of independent TPS over the internet.

A TV Broker server (TBS) provides Web Service to the clients, residing at the front end of the total system and controls the network integrity in the background, keeping record of the dynamic service model. The Internet TV Client (ITC) will get the service either from the most proximate TPS or another ITC receiving the service – the one that is more optimum, determined dynamically by the TBS. If several clients under a subnet request same service, a multicast network inside that subnet will be formed to deliver the service.

The paper is organized as follows: Section II contains several established architectures for internet television along with their draw backs. Section III contains the architecture of the system along with the detailed structure of the system and sequence of actions. We conclude the paper by pointing out some future research proposals.

II. BACKGROUND

Several protocols and backbone standards are defined for transmitting media over internet. RTP [3], [4] is a standard protocol to transmit streaming media. It provides the methods for transmitting media over networks and it is an accepted standard for Internet Radio, Internet Telephony, Music-on-Demand, Video-on-Demand, video conferencing and other multimedia technologies. MBone [9] is the multicast backbone to enable routers to multicast packets over internet. The Internet TV system could be based on MBone for content delivery. But MBone is an expensive specialized solution requiring external setup, which is not reachable and accessible for all clients over the internet. So it is not always appropriate for an Internet TV to be based on MBone.

Yima [8] defines a way to implement second generation continuous media server. But it cannot be used over current network systems and it requires expensive hardware.

RealNetworks and Unreal Streaming Technologies define proprietary technology for the management and delivery of media over internet. Their research works [5], [7] are in their early stages and the details are very vague.

III. PROPOSED ARCHITECTURE

A. SYSTEM COMPONENTS

In this section we describe the Service Architecture of the model for transmitting TV program in the Internet. There are two major modules in this model. The first module describes the various physical components of the system and the second module shows the inter-operation and interaction of these components. The proposed architecture for the Internet TV System consists of three service components:

- TV Broker Server (TBS)
- TV Program Server (TPS)
- Internet TV Client (ITC)

A.1. TV Broker Server (TBS)

TBS is the broker server for web service. It is also the controller which takes decisions about the interaction of the remaining part of the system dynamically. When the system is running, its main operations are threefold:

- It always communicates with the TV Program Servers and sends control messages and receives feedback.
- The client systems also interact with the TBS continually to receive information and policy about service and to change service status. TBS initiates and controls the interaction and service of a single client with other clients.
- All the transactions are internally represented and saved by the broker server as a Service Tree (ST) which is updated and maintained continuously. Based on the requests and feedback from client systems, TBS may decide to reorganize the network or some part of the network according to the approximate distance between the nodes (clients) recorded in the service tree.

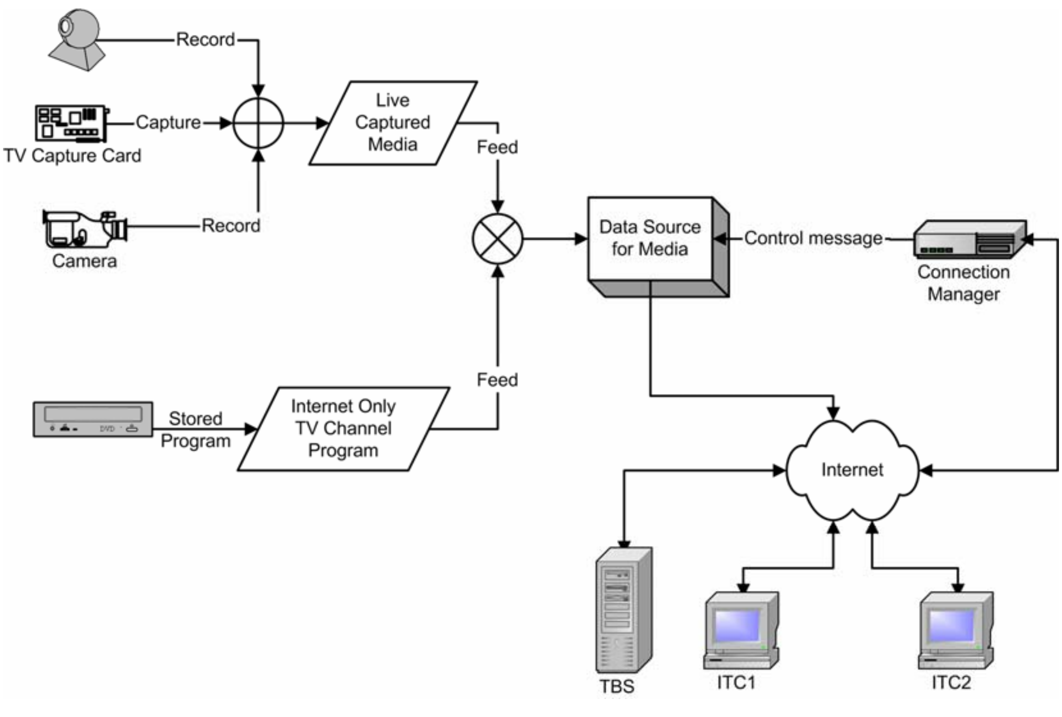

Fig. 1: A design for the TV Program Server and its relation with the other components

A.2. TV Program Server (TPS)

TPS is actually a cluster of servers rather than a single server. These servers capture live TV programs in real time, and they are capable of transmitting them to a client. TV programs will be captured by means of TV capture cards or similar devices. Customized schedules can be provided to set up low cost Internet-Only TV Channels (IOTC). Each TPS registers its service to the TBS, so that the TBS is aware of the actual TV service providers. When a TPS have to serve some client, it gets a command message from the TBS and then transmission starts.

A.3. Internet TV Client

ITC is the client system to watch TV programs. At first, it negotiates with the TBS, gets information about the service, receives service and sends feedback. During the total session it keeps connection with the TBS and waits for control messages from it.

An ITC in turn serves at most one client, and this serving is dynamic and controlled by the TBS. When more than one ITC in a LAN request same channel, one of the ITCs has to set up multicast group in that network as informed by the TBS. TV Service is then provided by that ITC and thus the bandwidth usage is balanced. Once the ITC that is directly getting the service from outside the LAN, is disconnected, another ITC in that multicast group gets direct service from outside and continues the multicasting. TBS is informed about each ITC in a LAN, but media transmission in the multicast group is handled by one ITC.

B. SERVICE OPERATIONS

B.1. Communication Setup

The typical communication scenario among the components is depicted in *figure 2*. In the center of the architecture we have a set of TPS (TV Program Servers).

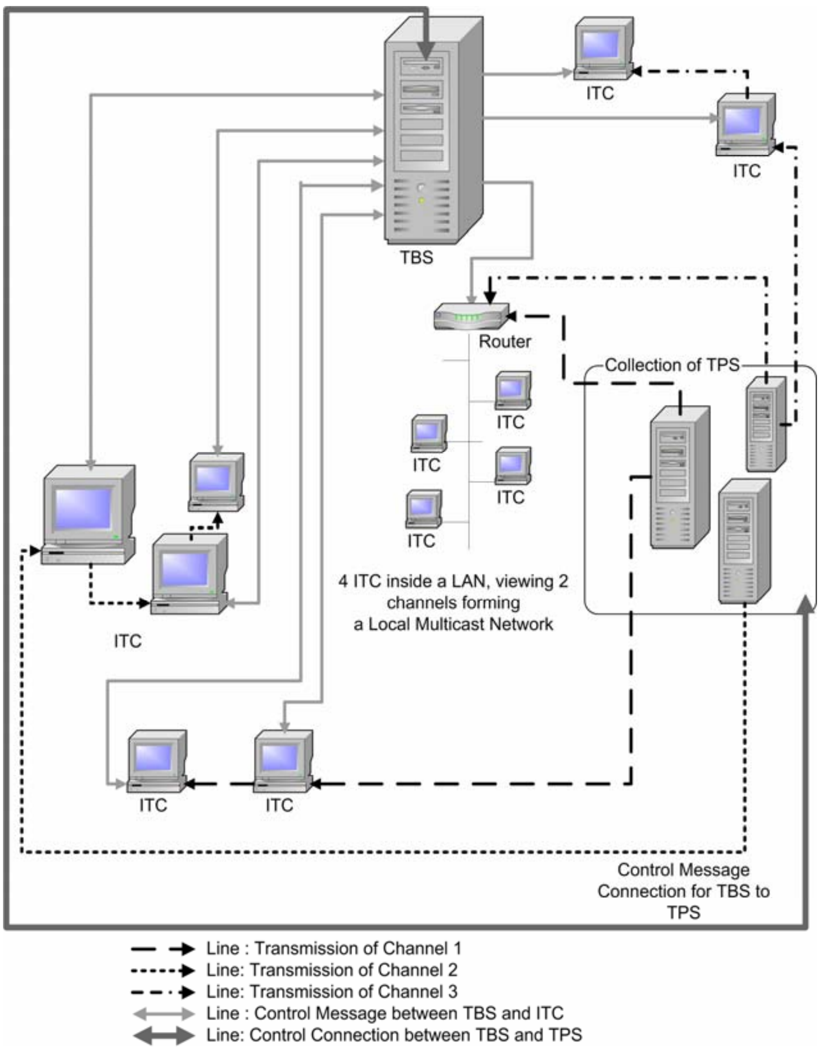

Fig. 2: Communication setup among TBS, TPS and ITC

These are actually the content deliverer in the Internet. Each TPS preserves TV programs of a specific channel. These TPS are scattered over the globe and are usually owned by some big companies that deploy the corresponding TV channel. There is a TBS (TV Broker Server) that acts as Service Registry in the model. Each ITC, which is actually the end user of the system, requests the TBS for desired TV channel. ITC gets their TV program from any TV Program Server (TPS) or any ITC that is currently getting the same TV channel, as determined by the TBS.

In *figure 2* TPSs are shown in a cluster, though they may be far apart, and do not need to have any connection between themselves. Transmissions of three hypothetical TV channels are shown as dotted lines. Any change in the service is decided by the TBS and the command is sent to ITC via black lines and to TPS via the bold gray lines.

B.2. Sequence of Action

TBS, TPS, and ITC have parallel actions dependent on one another. TBS takes the first actions and dynamically discovers the existence of the registered TPS over internet. After that, the system gets ready to work.

Service starts as an ITC requests TBS for service. TBS now performs its broker action, and, via the control connection sends message to some TPS to serve that ITC. TBS also updates its internal service tree for future calculation. As more requests arrive, TBS processes the service tree and makes a locally optimal choice about the ITC or the TPS that will serve a new client.

In this way, TBS always waits for request and controls the network in the background. Several situations may occur which will be handled by TBS. An ITC may be unable to serve another efficiently, or it may quit or wish to change the channel. In these cases, necessary decisions are taken by the TBS. In *figure 3*, a typical sequence of action is depicted. Other than serving and receiving, TPS and ITC also listen for control message from TBS. TBS may choose to change the network interconnection (reorganizing the network) periodically. Commands are then sent to respective ITCs and TPSs to

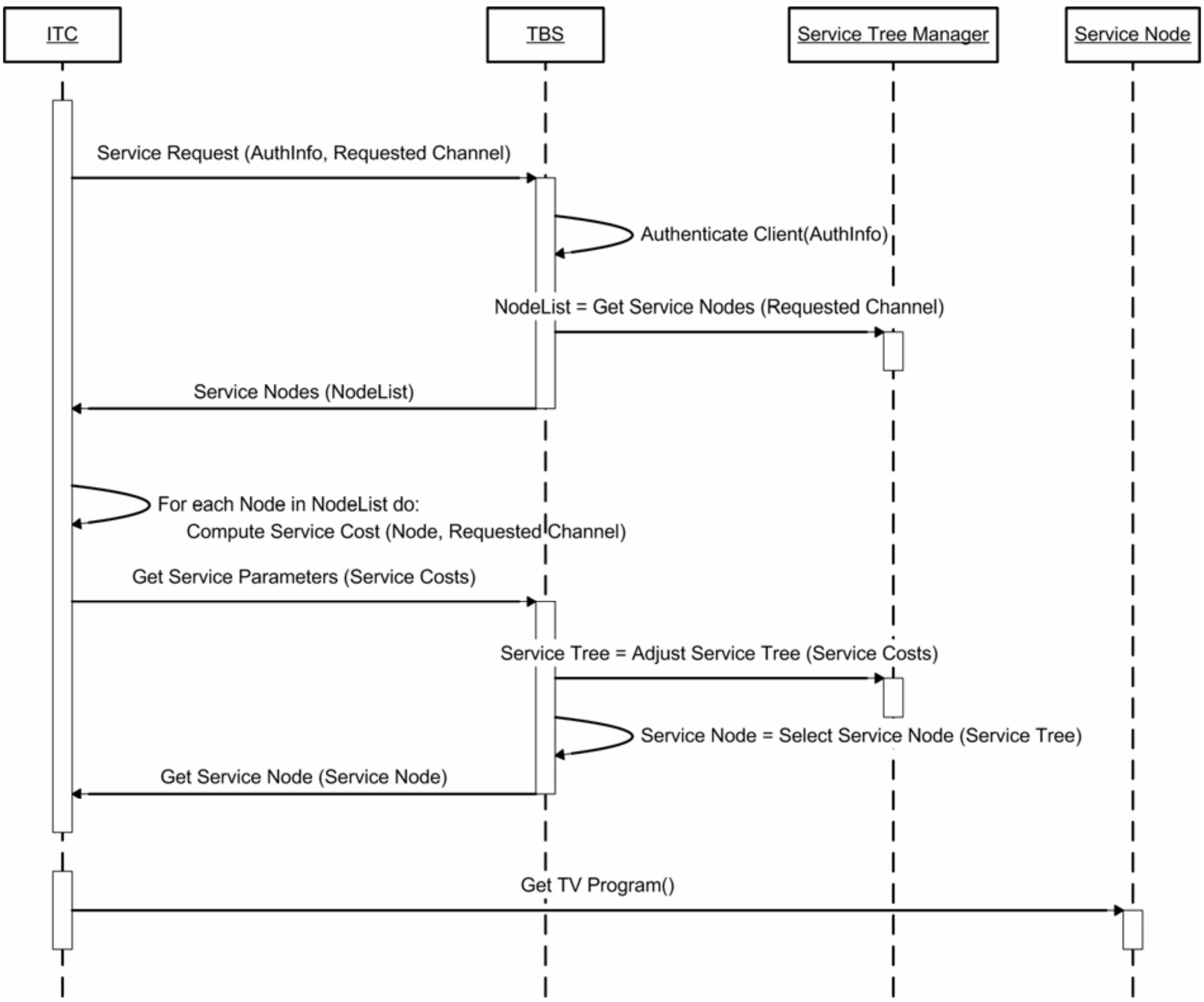

Fig. 3: Sequence of Action

apply the changes.

B.3. Service Tree Management

TBS always maintains a service tree for each channel as service registry. Service Tree (ST) contains the current status of the service. It shows the hierarchical involvement of TPSs and ITCs in receiving and providing TV programs for a channel. For each channel, there is a separate service tree.

TV Program Servers are at the root of each tree. These are the actual content providers. When a client gets service directly from a TPS, the ST is updated by connecting a node to a TPS. In the figure, there are three hypothetical TPS denoted by ***A, B, C*** and several ITCs denoted by small letters. The nodes are placed radial according to their relative distance and distance with the TPS. The distance here is network response distance, which may not be proportional to geographic distance all the time.

In *figure 4(a)*, TPS – ***A*** is directly serving ITC – ***a, b, c*** and TPS – ***B*** is serving ***d***. TPS – ***C*** is idle. Now we suppose that three new ITC – ***e, f, g*** are confirmed by TBS and they request channel ***A, B, C*** respectively. TBS verifies the tree, requests ***e*** to measure network response time with ***a*** and ***b*** and based on the feedback, finds that it is convenient to let ***b*** serve ***e***. TBS sends control message to both ITC and finally ***e*** is connected from ***b*** in the tree model (*figure 4(b)*). In a similar way ITC–***f*** is connected to ITC–***d***, and, ITC–***g*** is directly connected to ***C***.

Meanwhile, ***b*** now decides to view channel ***C***. So ***b*** cannot serve ***e*** anymore. TBS decides to serve ***b*** directly from TPS – ***C*** as there is no convenient ITC viewing channel ***C*** nearby. A control message is sent to ITC – ***a*** to serve ***e***. Also, a new node ***h*** appears to view channel ***B***. Contents are delivered to it from ***f***. The tree after these operations is shown in the *figure 4(c)*. Based on the feedback on the transmission quality or at some regular intervals, TBS may verify the entire tree and reorganize the network applying some change in the connection hierarchy to decrease network pressure and congestion. In *figure 4(d)*, one such reorganization is done. ITC – ***d*** now serves ***g*** and ***g*** serves ***f***. Moreover, as ***b*** decides to quit, it is deleted from the diagram.

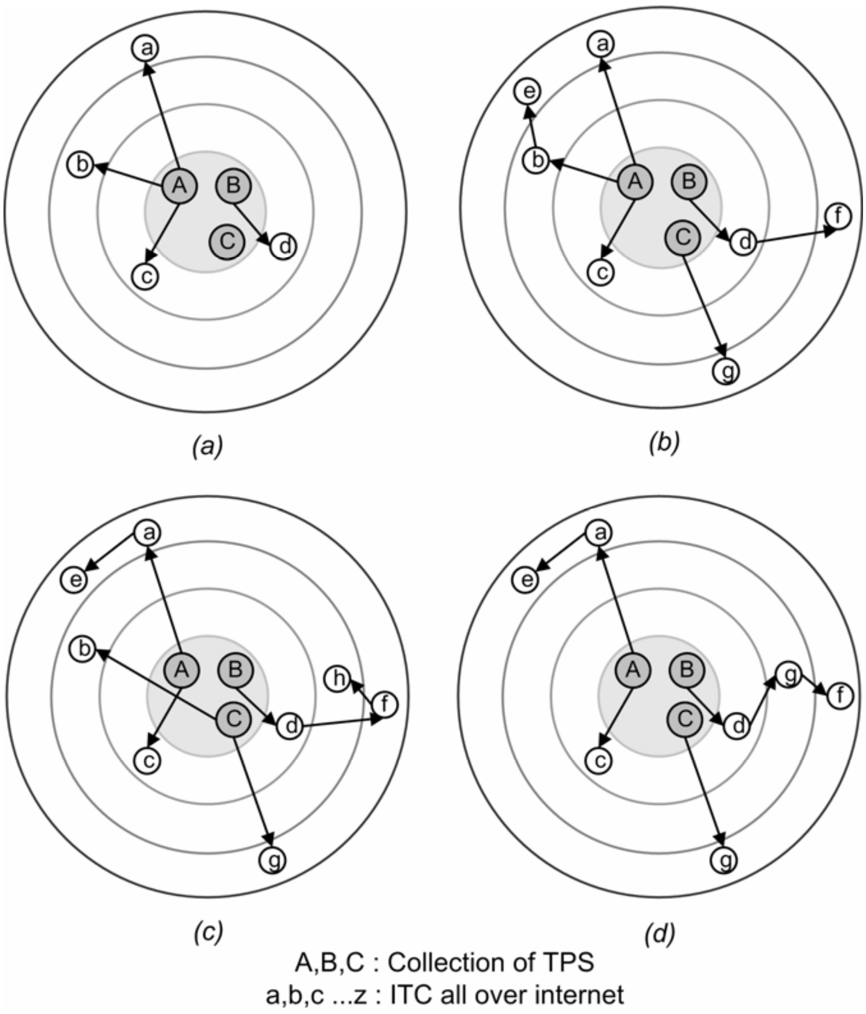

Fig. 4: Service Tree Management

IV. FUTURE WORKS

Simulating such a system and testing its performance in its full strength is one of our future goals. Present research handles with only the service model and content delivery. Maintenance and security of such a distributed system is a challenge. The system needs to have tight protocol between the TBS, TPS and ITC. Transmission can be controlled by RTP. But for better security, the Broker Server should itself control the transmission via the control and feedback connections. Future research work on this topic will include synchronization issues and overall security to prevent an unauthorized system from sabotaging the TV Network.

V. CONCLUSION

The main power of the proposed architecture is that it can be used over the existing network without any sort of special hardware requirement. The complexity of the existing architectures like MBone and Yima can be reduced to a great extent using this service oriented architecture. This will reduce the cost of service, and, the clients will be able to enjoy more real time television channels over the internet. In terms of scalability, interoperability and robustness the proposed Service Oriented Architecture seems to be perfect for Internet TV Systems.

REFERENCES

[1] A. S. Tanenbaum, *Computer Networks*. Pearson Education, October, 2002, pp. 704–714.

[2] G. Alonso, F. Casati, H. Kuno and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. Springer-Verlag, 2004, pp. 123–148.

[3] *RTP: A Transport Protocol for Real-Time Applications*. Network Working Group, RFC3550, July, 2003.

[4] *RTP Profile for Audio and Video Conferences with Minimal Control*. Network Working Group, RFC3551, July, 2003.

[5] *Internet Group Management Protocol, Version 3*. Network Working Group, RFC 3376, October, 2002.

[6] *IP Multicast Applications: Challenges and Solutions*. Network Working Group, RFC 3170, September, 2001.

[7] *Unreal Media Server*. Unreal Streaming Technologies. http://www.umediaserver.net/architecture.html

[8] C. Shahabi, R. Zimmermann, K. Fu and D. Yao, "Yima: A Second-Generation Continuous Media Server," IEEE Computer, June, 2002, pp. 56 – 64.

[9] K. Savetz, N. Randall and Y. Lepage, *MBONE: Multicasting Tomorrow's Internet.* Hungry Minds Inc, March, 1996.