Trust and Publishing Management for Streaming Media Services

Alsharif M. Ahmed and Qian Depei

***Abstract*—recently, networks are extended in multi-domains with scalabilities. The streaming media services require the dynamic trust management and maintain systems in multi-domains. This paper proposes a trust and publishing management method for streaming media services. The method dynamically maintains trust links for streaming media servers, applications and mobile nodes. It also integrates reputation authentication, forwarded authentication, streaming media publishing, and streaming media evaluation in one system, and improves the performance of trust, reliability and media publishing.**

***Index Terms*—Trust management, streaming media service, reputation authentication, evaluation.**

I. Introduction

Webs and IPTV publish their videos over Internet. Streaming media services are coming to our lives [1], but most of videos are transmitted over unsecured and unreliable networks. Some videos are directly transmitted from servers to users without any authentications, authorizations, evaluations and reliability [2]. Streaming media servers can't confirm whether their streams reach the destinations or not. There are no mechanisms for evaluating the service quality which media servers provided [3]. However, the streaming media requires some mechanisms and methods for the trust management, especially for media payment systems [4]-[6].

Streaming media services have their own characteristics. Traditional streaming media services publish videos from the source media server to clients. They provide users media streaming services with high bandwidth requirements. In general, VOD (Video on Demand) services needs at least 2Mbit/s bandwidth [7]. However, the existing user access bandwidth is hard to meet their requirements.

Most of clients access the streaming media directly from server. The streaming media which is transmitted from server to clients has the same quality and doesn't consider what kind of network they go [8]. Now, most of Webs and media servers use content distribution networks (CDNs) to distribute their media streams [9]. CDN is built on top of existing IP network infrastructure as a value-added network, deployed at the application, is a layer of network architecture.

CDN's core strategy is to use intelligent technology to push content and services from the center to network edges, enabling users and services in the nearest place to get the best quality service. CDNs dispatch the streaming media to different servers according to the multicasting tree. The key technologies of CDN, including caching, load balancing, content routing, content distribution, content storage and content management. CDN technology benefits include, global load balancing, access speed improvements, allowing users to connect to the nearest servers.

This paper proposes a method which manages the trust and publishing for the streaming media services.

Manuscript received August 5, 2012; revised September 14, 2012.

Alsharif M. Ahmed is with the School of Computer Science & Engineering, Beihang University (BUAA), 37 Xueyuan Road,,Haidian District Beijing ,China (e-mail: sharif_younis@yahoo,com).

Q. Depei is with Sino-German Joint Software institute Beihang University (BUAA), 37 Xueyuan Road Beijing China (depeiq@buaa.edu.cn).

II. Procedure for Paper Submission Streaming Media Publishing Management

The publishing of streaming media consists of two phases. Fig. 1 illustrates the streaming media publishing in Internet.

A. First Phase

In the first phase, the streaming media is published by media servers in the backbone. Media servers are linked as a tree, where the root media server distributes the streaming media to internal media servers and local media servers. Internal media servers catch the streaming media from the root media server or other internal media servers and forward to local media servers.

B. Second Phase

In the second phase, local media servers forward the streaming media to applications (e.g. media players). In this case, local media servers organize the local domain multicasting (LDM). LDM require the support of routers in the local domain. Applications catch the streaming media from local media servers and play it.

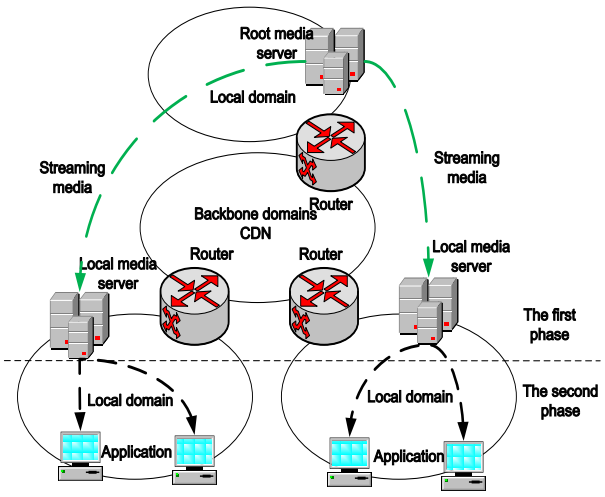

Fig. 1. Media publishing in the backbone and local domains

DOI: 10.7763/IJIMT.2012.V3.285

As shown in Fig.1, the root media server distributes two streams to two local media servers in local domains. This distribution is a multicasting service at the application layer. When streams reach local media servers, local media servers directly forward streams to applications in local domains, where some Ethernets support the low level multicasting. Comparing with traditional client-server distribution, the method is more like the multicasting service, which can improve the performance of distributions.

III. Authentication and Authorization for Media Servers

The CDN streaming media service is a good alternative to the media multicasting in Internet [10]. CDN establish the backbone multicasting tree among media servers. We designed the dynamic DNS (domain name service) for CDN streaming media services.

A. Dynamic DNS for CDN Streaming Media Services

Table I illustrates the dynamic DNS table, which forwards user requests to local media servers. In this table, we can find that applications access local media servers instead of the root server according to the source IP address in requests. CDN servers tell applications which local media server they can access and which publishing point they can catch and play as well. Moreover, CDN tell media servers how to find their father media server and child media servers in the multicasting tree.

For example the application from local domain "192.168.1.0"can find that its local media server is "192.168.1.10" and its publishing point is "video". Media server "192.168.1.10" can find its father media server is "192.168.10.100" and Media server "192.168.10.100" can find its child media servers "192.168.1.10" and "192.168.2.20".

TABLE I: Dynamic DNS for CDN Streaming Media Services

Local domain	Local media server	Publishing point	Father media server	Resolution
192.168.1.0	192.168.1.10	video	192.168.10.100	320×240
192.168.2.0	192.168.2.20	svideo	192.168.10.100	160×120
192.168.3.0	192.168.3.2	video2	192.168.10.10	240×180
192.168.4.0	192.168.4.10	video	192.168.10.10	320×240
...	...	...	…	...

B. Authentication and Authorization for Media Servers and Applications

Authentication and authorization for media servers and for applications are necessary, especially streaming media contents are related to payment systems, evaluation systems and reliability systems. Most of authentication and authorization methods in streaming media services mainly focus on two parts, Reputation authentication and authorization for media servers and authentication and authorization for applications.

1) Reputation authentication and authorization for media servers. All streaming media contents (files or live media) come from the root media server, so the internal media servers and local media servers must be authenticated and authorized by the root media server. Although all internal media servers and local media servers can directly communicate with the root media server, the root media server is fully overloaded. Moreover, the root server is not directly forwards streaming media to all internal media servers and local media servers. Hence, reputation authentication and authorization are required. The reputation authentication and authorization is designed as that the father media servers can authenticate and authorize the child media servers. Authenticated and authorized media servers get a copy of server register table, by which media servers could authenticate and authorize their child media servers.
2) Authentication and authorization for applications (users). Media servers not only transmit the streaming media to applications but also have the duty to authenticate and authorize applications. Local media servers manage and control applications in the local domains, and ensure that applications access the streaming media legally.

Fig. 2 illustrates the authentication and authorization tree for media servers and applications, where the root media server authenticates and authorizes internal media servers and internal media servers authenticate and authorize local media servers. Authorized media servers get a global server register table, which contain the information about usernames and passwords.

Authenticated media servers are authorized to authenticate and authorize applications. Applications would like to register their information to local media servers for authorized access.

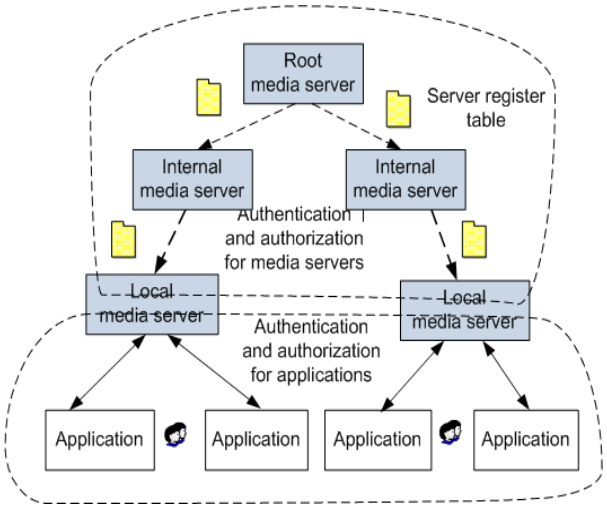

Fig. 2. Authentication and authorization tree for media servers and applications

IV. Forwarded Authentication for Mobile Nodes

Applications are locally managed and controlled by local media servers and the registered information from applications is not known to all local media servers. Broadcasting all authentication information to all media servers is impossible and not secure, so it is not easy for applications at mobile nodes to be authenticated and authorized in foreign domains.

We designed an authentication method for mobile nodes, called forwarded authentication. Fig.3 illustrates the forwarded authentication for mobile nodes. When the mobile node moves to a foreign domain, it sends the authentication information to local media server in the foreign domain. This local media server forwards the mobile node's authentication information to the media server in mobile-node's home domain for the authentication. If the authentication is succeeded, the local media server authorized the mobile node to access the streaming media. The forwarded authentication is easy for local media servers manage and control mobile nodes and applications.

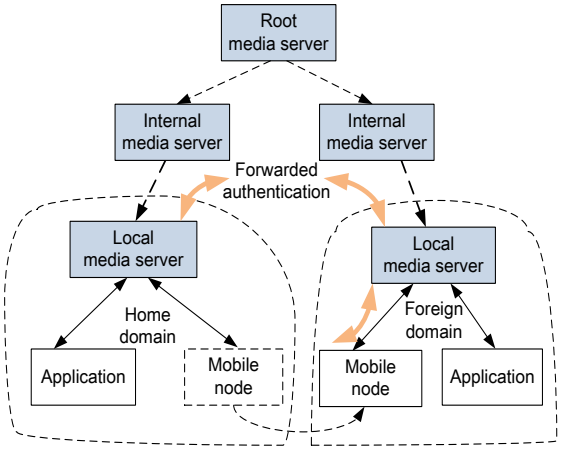

Fig. 3. Forwarded authentication for mobile nodes

V. Authentication and Authorization Based on HTTP

Most of authentication and authorization protocols use the HTTP in Internet. Webs get and treat the authentication information from applications, then response to applications via HTTP. Fig. 4 shows the authentication and access control.

The application issues HTTP requests to the authentication Web, which authorizes the application to access the media server. We use Microsoft Media Server (MMS) [11] as the streaming media server. MMS is designed to catch the streaming media from the root server or internal media servers, and push the streaming media to applications. MMS is also a name of the protocol to access Microsoft Media Server, so we write mms as the MMS protocol. Mms (e.g. mms://192.168.1.10/video) can access MMS publishing points, which push the streaming media to applications.

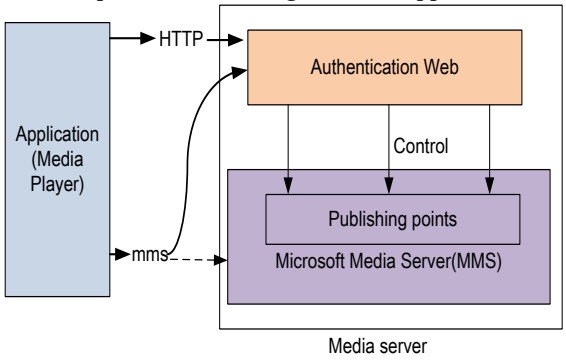

Fig. 4. Authentication and access control

Fig. 5 illustrates the implement of the authentication via HTTP in an application. The application issues the request, which includes the "name" and "password" pair for authentication, opens a HTTP connection to the authentication Web, and waits for the response from the authentication Web. After the application gets the response (e.g. mms://192.168.1.10/video, as shown in Table 1 with specified resolution), it access the publishing point.

```
url = new URL("http", HostName, mdsPort, "/
      authenticate?name=movie&password=movie");
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestProperty("Content-Type","text/xml;charset=utf-8");
conn.setDoInput(true); // for receiving the confirmation
conn.setDoOutput(true); // for sending the data
conn.setRequestMethod("POST"); // post the data to the authentication Web
OutputStream out = conn.getOutputStream();
out.write(data); // write the data
out.close();
conn.getResponseCode();
conn.disconnect();
```

Fig. 5. Authentication via HTTP

The authentication and authorization based on HTTP is easy to issue request, which can be forwarded to any media servers. Hence, media servers provide unified interfaces for traditional authentications, reputation authentications and forwarded authentications.

VI. Streaming Media Reliability and Classification

Local domains have various communications capacities, 2Mbps or 100Mbps, so they can't meet the requirement of different media qualities. Streaming media becomes less reliability to applications and not trustable to users.

Fortunately, MMS provides various video resolution publishing points for applications in order to meet different media quality requirements. The method for streaming media evaluations and classifications is shown in the Fig.6.

1) The method classifies various streaming media contents, which have specified video resolutions (e.g. 320×240 and 240×180), into different publishing points.

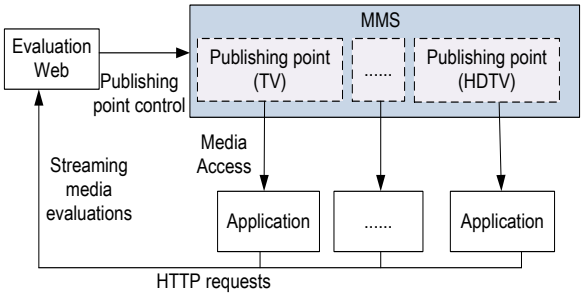

Fig. 6. Streaming media evaluations and classifications

2) Applications (e.g. Microsoft Media Player) play the streaming media from publishing points. As applications find that the resolutions are not suitable for them, they feedback to the evaluation Web, which controls MMS's publishing points. The evaluation Web evaluates the responses and choice proper publishing points for applications to ensure the reliability.

The resolution is calculated and evaluated by the streaming buffer. Suppose the data buffering speed is V_i bit/second. The video resolution requires that the buffering speed should be more than R bit/second. The proposed video resolution should meet the following form:

$$R \leq \sum_{i=0}^{n} V_i / n \quad (1)$$

Fig.7. shows application's feedback to the evaluation Web via HTTP. Applications issue the HTTP request to the evaluation Web. The request in HTTP includes some parameters related to the video qualities, such as the current video resolution (old resolution = 320*240) and proposed video resolution (new resolution = 240*180). The evaluation Web gets the request, finds a MMS publishing point according to the request, then responses the publishing point to the application via HTTP.

```
url = new URL("http", HostName, mdsPort, "/
      evaluate?oldresolution=320*240&newresolution=240*180,...");
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestProperty("Content-Type","text/xml;charset=utf-8");
conn.setDoInput(true); // for receiving the confirmation
conn.setDoOutput(true); // for sending the data
conn.setRequestMethod("POST"); // post the data to the evaluation Web
OutputStream out = conn.getOutputStream();
out.write(data); // write the data
out.close();
conn.getResponseCode();
conn.disconnect();
```

Fig. 7. Application's feedback to the evaluation web via HTTP

VII. Conclusion

The streaming media publishing use CDN services and two-phase distribution services improves the performance of publishing, which like media multicasting, comparing with the traditional media stream transmission. The reputation authentication and authorization method for media servers, which applies trust management in the system, enhances the performance of dynamic and real time services comparing with the traditional client-server authentications and authorizations.

In local domain, local media servers forward streams directly to or multicast to applications according to the evaluation of stream qualities. The evaluation helps media servers to control the video resolutions. The evaluation method improves the reliability of publishing comparing with the unaware client-server publishing.

Acknowledgment

I am thankful for Libyan Ministry of Higher Education & Scientific Research for providing me this outstanding opportunity to undertake my studies at Beihang University (BUAA) in China.

My deep gratitude and thanks to my advisor Prof. Dr. Qian Depei, he taught me to work hard is the way to succeed. He guided me and encouraged me to develop my skills as researcher. There are no words to express my deepest sense of love for my parents, without them I would not be here, thanks for their love, support, dedication, and encouragement.

References

[1] M. H. Hayes, "Some approaches to Internet distance learning with streaming media," *IEEE Second Workshop on Multimedia Signal Processing*, pp. 514-519, Dec 1998.

[2] J. G. Apostolopoulos and M. D. Trott, "Path diversity for enhanced media streaming," *IEEE Communications Magazine*, vol. 42, no. 8, pp. 80-87, Aug 2004.

[3] M. Hicks and A. Nagarajan, "User-specified adaptive scheduling in a streaming media network," *IEEE Conference on Open Architectures and Network Programming*, pp. 87-96, April 2003.

[4] Anwitaman Datta, Manfred Hauswirth, and Karl Aberer, "Beyond "web of trust": Enabling P2P E-commerce," in *Proc. the IEEE International Conference on E-Comerce (CEC'03)*, 2003.

[5] D. McKnight and N. Chevany, "The Meanings of Trust," *Working paper*, Carlson School of Management, University of Minnesota, 1996.

[6] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proc. the 1996 IEEE Symposium on Security and Privacy*, pp.164-173, May 1996.

[7] Liu Ya-Jie and Dou Wen-Hua, "A Video-on-Demand Streaming Service Architecture in P2P Environment," *Journal of Software*, no. 4, 2006.

[8] Venkata N. Padmanabhan, Helen J. Wang, Philip A.Chou, and Kunwadee Sripanidkulchai, "Distributing Streaming Media Content Using Cooperative Networking," *Microsoft Technical Report*, April 2002.

[9] J. Kangasharju, J. W. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," in *Proc. the 6th Web Caching and Content Distribution Workshop*, (Boston, MA), June 2001.

[10] Eveline Veloso and Virgilio Almeida, "A hierarchical characterization of a live streaming media workload," in *Proc. the 2nd ACM SIGCOMM Workshop on Internet measurement*, 2002.

[1] Microsoft Corporation, Windows Media Services. (2011). [Online]. Available: http://www.microsoft.com/windows/windowsmedia/forpros/server/server.aspx.

Alsharif M. Ahmed was born in July 22 1970. He received the Bachelor's degree in Computer science & Engineering in Computer Security at "Engineering Academy Tajoura" Tripoli Libya since 1993. He was enrolled as a Master candidate in the School of Computer Science & Engineering "Computer Architecture & Networking" at Beihang(BUAA) University in March 1999, Beijing P.R of China and Graduated in March, 2003. He is currently a PhD candidate at Beihang University (BUAA) since May 2008.

He holds the HEAD of computer department In the Consultative Office for the Utilities in Tripoli/Libya1995-2000, EXECUTIVE DIRECTOR of projects & Constructions in Administration of Engineering Department field 2006-2009.His current research work focused on (Trust Management for Collaboration in Network).

Qian Depei was born in Shanghai, China, August 1952. He graduated from Xi'an Jiaotong University in 1977 computer professional, a master's degree from North Texas State University in the U.S. state of Texas in May 1984. June 1991 to 92 March as a senior visiting scholar of Computer Science, University of Hannover, Germany, the system structure and education work of the Institute of the operating system. Professor since 1992 was hired as a doctoral supervisor in 1996. Since 1996, members of the Expert Group on National 863 Program 306 theme, deputy head of the current national "Eleventh Five-Year Plan 863 IT members of the Group in the field, 863 high-performance computer and grid service environment overall project group long, the Chinese Computer Society. Since 1996 he has been involved in the activities of the expert group for the National High-tech Research & Development Program (the 863 program).

Prof. Depei, DIRECTOR of the Sino-German Joint Software Institute at Beihang University (BUAA), He has been working on computer architecture and computer networks for many years.