ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

씬클라이언트 환경에서 클라우드 컴퓨팅을 이용한 N-Screen 세션 관리 기반의 N-Screen 서비스 프레임워크

A Framework of N-Screen Session Manager based N-Screen Service using Cloud Computing in Thin-Client Environment

아이만 압둘라 알사파르* 송 표** 모하마드 메히디 핫싼*** 허 의 남****
Aymen Abdullah Alsaffar Song Biao Mohammad Mehedi Hassan Eui-Nam Huh

요 약

본 논문은 씬클라이언트 환경에서 클라우드 컴퓨팅을 이용하여 N-Screen-as-a-Service(NaaS) 기반하여 통합 된 어플리케이션을 스트리밍하기 위한 Virtual Aggregation Gateway(VAG)를 제안한다. 또한, 본 논문에서는 인터넷을 통하여 대용량 멀티 클라이언트에서의 융합된 어플리케이션 스트림의 화면 공유 시 서버 측 문제에 대해 기술한다. 특히, 콘텐츠를 전달하기 위해 가능한 모든 미디어 시그널링을 효율적으로 관리를 하는 N-Screen 세션 관리자 프레임워크를 제안한다. 더 나아가, 사용자들에게 재생 멀티미디어 콘텐츠 (TV 드라마, 광고, 대본 등)를 제공한다. 제안한 N-Screen 세션 관리자의 목표는 다음과 같다. (1) 모든 통신 세션의 세션 상태를 관리 (2) 수신된 요구사항이나 답변들의 제어를 관리 (3) 사용자들에게 언제 어디서든 이기종 단말에 화면을 공유하면서 멀티미디어 콘텐츠를 재생 그리고 (4) 사용자들이 진행 중인 통신 세션을 이기종 단말로 전환을 가능하게 해준다. 마지막으로, Session Initiation Protocol (SIP)에서의 보안 문제점 및 재생 혹은 전송 세션에서의 지연을 최소화 하는 것에 대해 기술한다.

ABSTRACT

We develop architecture of a virtual aggregation gateway (VAG) which enables composite application streaming based on N-Screen-as-a-Service (NaaS) using cloud computing in thin-client environment. We also discuss the problem of server computing burden in large scale multi-client case for screens sharing with composite application streaming over the internet. In particular, we propose an efficient Framework of N-Screen Session Manager which manages all media signaling that are necessary to deliver demanded contents. Furthermore, it will provides user with playback multimedia contents method (TV Drama, Ads, and Dialog etc) which is not considered in other research papers. The objectives of proposing N-Screen Session Manager are to (1) manage session status of all communication sessions (2) manage handling of received request and replies (3) allow users to playback multimedia contents anytime with variety of devices for screen sharing and (4) allow users to transfer an ongoing communication session from one device to another. Furthermore, we discuss the major security issues that occur in Session Initiation Protocol as well as minimizing delay resulted from session initiations (playback or transfer session).

☞ keyword : N-Screen 세션 관리자(N-Screen Session Manager), 씬클라이언트(Thin-Client), 클라우드 컴퓨팅(Cloud Computing), SIP, 인증기법(Authentication)

1. INTRODUCTION

Now a days N-Screen-as-a-Service (NaaS) in cloud computing is gaining rapid popularity by users for the services the cloud can offer. It provides users with an easy access to contents or sharing screens across one or more devices that are Internet-connected to each other (e.g. PDA, Smart Phone, IPTV, Thin-Client, PC etc) in a secure, seamless and transparent way. In spite of that, the recent N-Screen service in cloud computing platform focus only on one application streaming scenario.

The Idea of deploying N-Screen Session Manager which enables users to playback contents from different servers to variety of N-screen devices using cloud computing in

* 정 회 원 : 경희대학교 컴퓨터공학과 박사
aymen@khu.ac.kr
** 정 회 원 : 경희대학교 컴퓨터공학과 박사
bsong@khu.ac.kr
*** 정 회 원 : 경희대학교 컴퓨터공학과 박사
hassan@khu.ac.kr
**** 종신회원 : 경희대학교 컴퓨터공학과 정교수
johnhuh@khu.ac.kr(교신저자)
[2011/12/10 투고 – 2011/12/15 심사 – 2012/01/17 심사완료]

http://dx.doi.org/10.7472/jksii.2012.13.2.21

thin-client environment was not considered. In addition N-Screen Session Manager responsible of managing session status of all communication sessions, managing received requests/replies, and allows users to transfer an ongoing communication session from one device to another. Session Initiation Protocol (SIP) is an application-layer control protocol that can create, modify, or terminate user's sessions [8].

These sessions includes multimedia conference, distance learning, internet telephony, multimedia distribution, and instance messaging applications, as well as many others [8]. SIP is widely used as a signaling protocol [9]. However, SIP system is deployed in the internet that can be considered hostile environment in which SIP elements and messages may be exposed to a variety of security threats and attacks (e.g. Replay attack, Request Spoofing, Impersonating a Server, etc) [8]. As a matter of fact, the deployment of this service is very useful for thin-client users because it is difficult for them to receive multiple services in one device.

Implementing this service is not easy. One of the greatest challenges faced by service providers revolve around 1) How to make this new service easy, enjoyable, compelling, flexible and cost effective. 2) How to make this service more secure against the previously mentioned security threats and attacks. 3) Minimizing delays resulted from session initiation (playback media, user authentication or session transferring). The user will likely to appreciate this service for having the feature of being secure, seamless and transparent.

In playback the multimedia contents over and over, 1) users should not have to continuously log in and log out of different devices 2) Users should not maintain and store multiple profiles for each device which they will use and 3) Users should not exhaustibly have to look for relevant multimedia contents. Therefore, we proposed a framework of N-Screen Session Manager which manages all users sessions based on N-Screen services using cloud computing for thin-client environment. Our proposed is designed to 1) manage session status of all communication sessions, 2) manage handling of received requests/replies 3) allow users to playback multimedia contents anytime with variety of devices for screen sharing and 4) allow a users to transfer an ongoing communication session from one device to another. As for the Security of SIP, several aspects must be taken into consideration. The exchange of messages between users and proxies will contains sensitive information (e.g. URLs, phone number, IP address, etc) that should be protected. There is a need to assure the identity of communicating users and integrity of messages as well. We will discuss some previous authentication mechanisms used with SIP.

The remainder of this paper is organized as follows: in section 2, we review some of the background. Section 3 presents basic system architecture of virtual aggregation gateway for N-Screen Services using Cloud Computing in Thin-Client Environment. Section 4 presents our proposed system architecture of N-Screen Session Manager based N-Screen Services using Cloud Computing in Thin-Client Environment as well as scenarios. Section 5 present Security Analysis and Discussions, and lastly, Section 6 present our Conclusion.

2. BACKGROUND

2.1 SESSION INITIATION PROTOCOL (SIP)

Session Initiation Protocol (SIP) is the Internet Engineering Task Force standard (IETF) for IP telephony. It is considered to be the most promising candidate for call setup signaling for future IP-based telephony services. Moreover, it has been chosen by the Third-Generation Partnership Project (3GPP) as the protocol for multimedia application in 3G mobile networks [10]. SIP is an application-layer control (signaling) protocol which is used for creating, modifying, and terminating sessions (e.g. Internet telephone calls, multimedia distribution, and multimedia conference) with one or more participants [1, 10]. The session between two user agent clients (UACs) is established using signaling mechanism which involves sending INVITE, OK response and ACK to the OK message [11].

SIP message contain a body that stores information related to the session establishment using a text-based representation called Session Description Protocol (SDP) [10]. SIP message may contain information a user or server wishes to keep private. The header can reveals information about the communication pattern and content of individuals or other confidential information (e.g. codec, media types, addresses and ports, etc.) that should not be revealed [10].

Some of SIP features are 1) Provides primitive that can be used to implements different services such as locating users and deliver an opaque object to his/her current location. In other word, a single primitive is used to provide several different services. 2) It can be used to initiates a session that uses some existing conference control protocol which make it more flexible. 3) Provides a suite of security services which include denial-of–services prevention, authentication for user to user and proxy to user, integrity protection, encryption, and privacy services [1].

2.2 N-SCREEN

N-Screen is composed by various devices such as web, mobile, and PCs. N-Screen services attract increasing attention between wired and wireless. Device is interlinked by various devices and communication network is developing which especially enables sharing of contents and services for various devices [2].

2.3 SIP DIGEST AUTHENTICATION

The origin of SIP digest authentication [9] in SIP was base on HTTP Digest mechanism [12]. It is known as challenge-response which is used as authentication for client to client or client to proxy. Here the recipients can challenge the sender identity using nonce value. To respond to this challenge, the sender sends a message digest calculated using nonce value, username, user password, realm and other optional parameters. There are five different headers for SIP Digest authentication. There are WWW-Authentication, Authorization, Proxy-Authorization, Proxy-Authenticate, and Authentication-Information. When SIP user agent send request, he/she needs to be authenticated. For example user agent sends messages with WWW-Authenticate header where the header contains nonce value, user name, user password, realm and other optional parameter. A nonce value is unique string generated every time "401 Unauthorized" message is sent and realm specifies the digest algorithm used for challenge [13]. After receiving challenge, client computes response value using nonce value, user name, user password, realm and other optional parameters which included in Authorization header in new request message. MD5 algorithm is used to compute this response value [13].

2.4 THIN-CLIENT

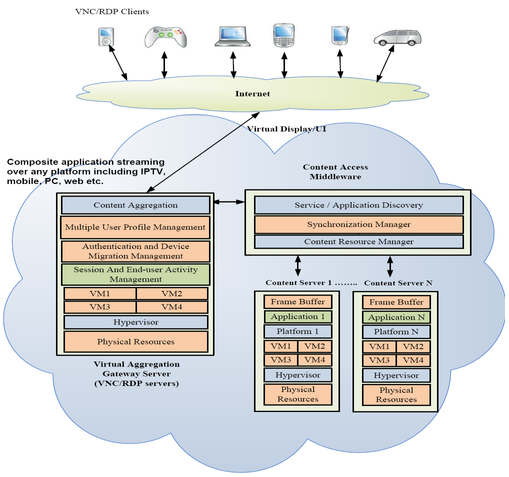
Thin-client is a computer or a computer program which depends heavily on some other computer such as servers to do its computational roles [3]. Thin-client platform consists of client applications that executes on a user local desktop machine and a server application that executes on a remote system. The end user machine can be a hardware device designed specifically to run the client applications or a low-end personal computer. The remote server machine typically runs a standard server operating systems, where client and server communicates across a network connection between the desktop and server. The client sends input data across the network to the server, and the server returns display updates [3].

2.5 CLOUD COMPUTING

A Cloud is a type of parallel and distributed system consist of collections of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service providers and consumers[4]. Cloud computing provides a shared pool of configurable IT resources (e.g. processing, network, software, information and storage) on demand, as a scalable and elastic service, through a networked infrastructure, on a measured (pay-per-use or subscription) basis, which needs minimal management effort, is based on service level agreements (SLA) between the service provider and consumers, and often utilizes virtualization resources. This frequently takes the form of web-based tools or applications that users can access and use through a web browser as if it was a program installed locally on their own computer [5]. Major advantages of the cloud computing are, SaaS (Software as a service), Utility Computing, Network service, PaaS (Platform as a service), MSP (management service provider), Commercial service platforms, and integrating internet [6]. Other advantages of cloud computing are 1) Provide faster, simpler and cheaper services. 2) It is highly elastic because resources are easily released or occupied on the basis of demands. 3) It optimizes utilization of computing resources. 4) Users have more resources, unlimited storage, and everything is provided as services [7].

3. BASIC SYSTEM ARCHITECTURE OF VIRTUAL AGGREGATION GATEWAY

Figure 1 illustrates the basic system architecture of virtual aggregation gateway for N-Screen services using clouding computing in thin-client environment. The terminologies used to describe the system architectures are explained below. The architecture has three parts as follows: virtual aggregation gateway server, content access middle ware, and content server.



(Figure 1) Basic System Architecture of Virtual Aggregation Gateway

Virtual Aggregation Gateway Server will contain the following components:

CONTENT AGGREGATION AND MANAGEMENT: responsible for creating composite application streaming and ensuring the integrity of the application streaming across multiple screens.

MULTIPLE USER PROFILE MANAGEMENT: responsible for managing consumer-specific profiles across multiple services delivery platforms with multiple devices.

AUTHENTICATION AND DEVICE MIGRATION MANAGEMENT: responsible for managing the authentication, reformatting, re-encoding, ingestion and correlation of contents for the various services delivery networks.

SESSION AND END USER ACTIVITY MANAGEMENT: responsible for managing different user's sessions on different devices as well as their current activities in those sessions.

SERVICE/APPLICATION DISCOVERY (SD): responsible to discover appropriate composite applications based on the users activities profiling and current contexts.

SYNCHRONIZATION MANAGER: responsible of maintaining the synchronization of contents/applications streaming received from different content servers for various user sessions in multiple thin-client devices.

CONTENTS RESOURCE MANAGER: responsible for allocating the VM resource necessary for running users applications.

CONTENT SERVER: responsible for host the user applications in different virtual machines with different platforms.

4. SYSTEM ARCHITECTURE OF N-SCREEN SESSION MANAGER

A. SCENARIOS

We considered two main scenarios. First scenario illustrated in figure 2, we are assuming that the user is using one of their devices (e.g. TV, PDA, PC, Smartphone, etc) which are capable of showing TV Drama and Ads in one screen at the same time. While the user is watching the ads, the TV Drama is already finished. Here the user would like to go back to watch the part of the ads or the TV Drama which they missed. The issue here is how to provide the user with easy and flexible way to go back to watch the missing part.

In order to split sessions across multiple devices, the

(Figure 2) First Scenario Idea

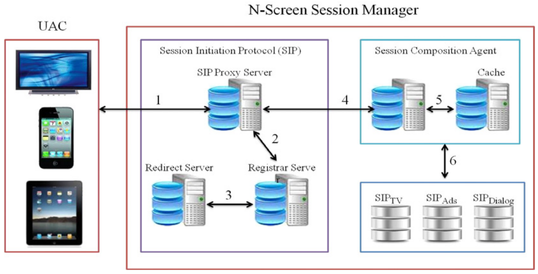
(Figure 3) Second Scenario Idea

UAC establishes a new session with other device through a separate INVITE request, and updates the existing session with SIP Proxy Server by an SDP body that combines appropriate media parameters which is received in their response [15].

In our second scenario illustrated in figure 3, we are assuming the user is watching both TV Drama and Ads in one screen. The user would like to have the flexibility to transfer one of the programs TV Drama or Ads to another device (terminal) according to the main contents. So the user will watch TV Drama in one device and Ads in another one. The issue here is how to deliver all user contents again to user different devices. This will required to split session in the original device and transfer it to the desired device.

B. SYSTEM ARCHITECTURE

Figure 4 illustrate system architecture of our proposed N-Screen Session Manager which consists of User Agent Client (UAC) and N-Screen Session Manager. N-Screen Session Manager consists of Session Initiation Protocol (SIP), Session composition agent, Cache, and SIP for TV Drama, Ads and Dialog. The terminologies used to describe the system architecture are illustrated below.



(Figure 4) System architecture of N-Screen Session Manager

(Table 1) SIP components and their role

User Agent Client (UAC):	It is the end-user devices, such as cell phone, multimedia handset, PC, and PDA. Responsible for creates and manages SIP sessions where UAC create requests and UAS respond to it.
SIP Proxy Server (SIP PS):	A proxy server forward requests and responses between a callee and caller. When proxy server receives a request, it will forward requests to current location of the callee, and then forward responses from the callee to caller.
Registrar Server (RS):	It is database that responsible for contain the location of all User Agents within a domain. Responsible for retrieve and send participants' IP address and other pertinent information to the SIP Proxy Server.
Redirect Server (RS):	Responsible for allowing Proxy Server to direct SIP session invitation to external domains.
Session Composition Agent (SCA):	Responsible of storing Session Identifications and Contents to Cache for streaming or playback. Responsible for composting all contents and transmits to users devices.
SIP for TV (SIPTV, SIPAds, SIP Dialog):	Responsible for creating, modifying, and terminating TV Drama, Ads and Dialog sessions.

1) Session Initiation Protocol (SIP).

SIP session utilize up to four major components: User Agent Client, SIP Proxy Server, Registrar Server, and Redirector Server. Together, these systems deliver messages embedded with the SDP protocol defining their contents and characteristics to complete a SIP session.

2) Session Composition Agent.

It utilizes up to 2 major components: Session Composition Agent and Cache. Table 1 provides description of each SIP component and the role it plays in this process.

C. SYSTEM SEQUENCE OF N-SCREEN SESSION MANAGER

Figure 5 illustrates sequence chart of N-Screen Session Manager first scenario. In order to describe figure 5 clearly, we provide description of the system parameters in table 2.

In our first scenario illustrated in figure 5, when the user request to access resources of the server, he/she proceeds with following steps:

(Table 2) System Parameters

Notation	Description
D_{ID1}	Identity of user 1st device.
D_{ID2}	Identity of user 2nd device.
D_{PW}	Password of user device.
D_{IP1}	IP address of user 1st device.
D_{IP2}	IP address of user 2nd device.
D_{SI1}	1st device session initiation.
D_{SI2}	2nd device session initiation.
D_{L}	Location of user device.
D_{nonce}	Nonce value of user device.
$SIPPS_{nonce}$	Nonce value of SIP Proxy server.
$D_{content1}$	Content of user 1st device.
$D_{content2}$	Content of user 2nd device.
E()	Encryption.
T_{SI}	Transfer session (TV , Ads, Dialog, etc).

Step1. UAC1 to SIP Proxy Server:

The user send request (INVITE) followed with {D_{ID1} | D_{SI1} | D_{nonce}} and device generated nonce value to (prevent replay attack) to SIP Proxy Server (SIPPS) to request for session initiation. In return, SIPPS will request {D_{ID1} | D_{IP1} | D_{PW1} | $D_{content1}$ | D_{nonce} | $SIPPS_{nonce}$} followed by SIPPS generated nonce value. When user authentication is validated by SIPPS, then request will be forward to Registrar Server.

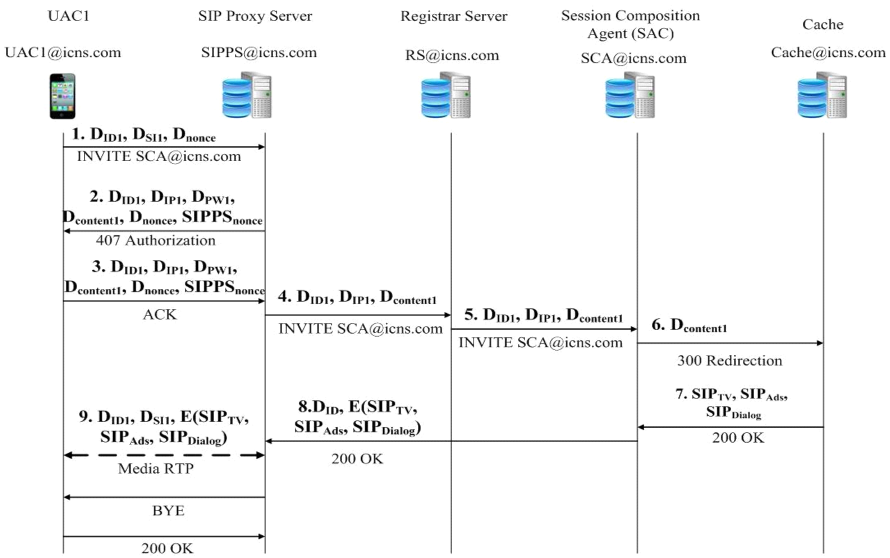
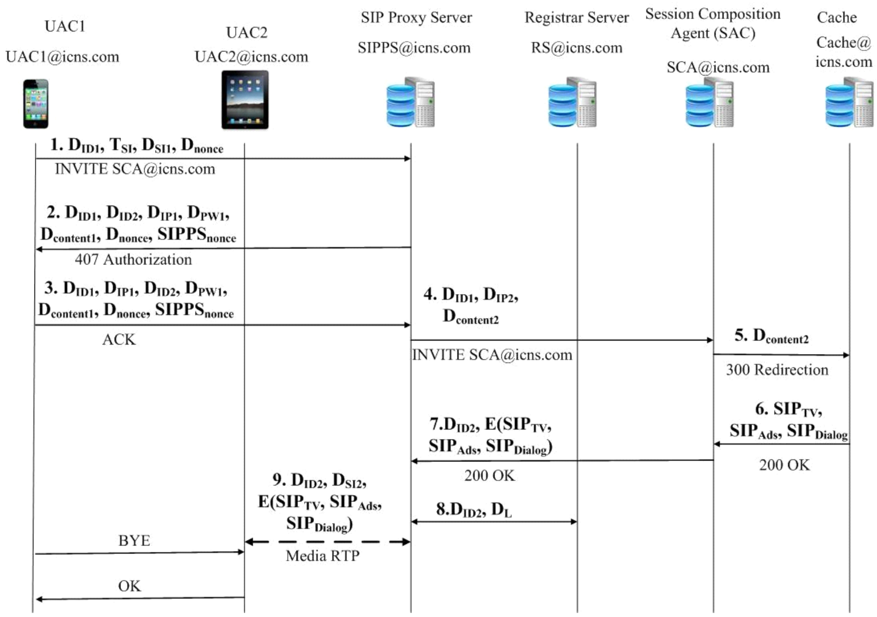


Figure 5 Sequence of N-Screen Session Manager First Scenario



(Figure 6) Sequence of N-Screen Session Manager Second Scenario

Step2. SIP Proxy Server to Registrar Server:

Upon the validation of user sent security information, SIP Proxy Server forwards user request {D_{ID1} | D_{IP1} | $D_{content1}$} to Registrar Server.

Step3. Registrar Server to Session Composition Agent:

Here it will store user information, retrieve user IP address, and other information {D_{ID1} | D_{IP1} | $D_{content1}$}. Finally, Registrar Server forwards the same request to Session Composition Agent.

Step4. Session Composition Agent (SCA) to SIP Proxy Server:

Here Session Composition Agent will redirect request to cache in order to retrieve the user desired contents. Then SCA will receive content from cache, prepare, encrypt and send to SIP Proxy server.

Step5. SIP Proxy Server to UAC1:

SIP Proxy Server receives user contents {D_{ID1} | E(SIP_{TV}, SIP_{Ads}, SIP_{Dialog})}. The SIP Proxy Server initiate the session by sending the following to user device {D_{ID1}| D_{SI1} | E(SIP_{TV}, SIP_{Ads}, SIP_{Dialog})}and this will finalize the first scenario steps.

In our second scenario in figure 6, when the user requests to transfer some of current watched contents to other devices, he/she proceeds with following steps:

Step1. UAC1 to SIP Proxy Server:

The user send request (INVITE) followed with {D_{ID1} | T_{SI} | D_{SI1} | D_{nonce}} and device generated nonce value to SIP Proxy Server (SIPPS) to request for transfer session initiation to other devices. In return, SIP PS will request {D_{ID1} | D_{ID2} | D_{PW1} | $D_{content1}$ | D_{nonce} | $SIPPS_{nonce}$} followed by SIPPS generated nonce value. When user authentication is validated by SIPPS, then request will be forward to Session Composition Agent.

Step2. SIP Proxy Server to Session Composition Agent:

Upon the validation of user sent information, SIP Proxy Server will forward user request {D_{ID2} | D_{IP2} | $D_{content2}$} to Session Composition Agent.

Step3. Session Composition Agent to SIP Proxy Server:

Here Session Composition Agent will redirect request to cache in order to retrieves the user desired contents {$D_{content2}$}. Then SCA will receive contents from cache, prepare, encrypt and send to SIP Proxy Server {D_{ID2} | E(SIP_{TV}, SIP_{Ads}, SIP_{Dialog})}.

Step4. SIP Proxy Server to Registrar Server:

SIP Proxy Serve will send {D_{ID2} | D_L} to Registrar Server will stores D_{ID2} address location and send it to SIP Proxy Server.

Step5. SIP Proxy Server to UAC2:

After receiving D_{ID2} address location, SIP Proxy Server will locate the device and send {D_{ID2} | D_{SI2} | E(SIP_{TV}, SIP_{Ads}, SIP_{Dialog})} which will establishes the session initiation for user second device.

5. SECURITY ANALYSIS AND DISCUSSION

What kind of security vulnerabilities exist and what are the threats according to SIP authentication. We analyze previous SIP authentication mechanisms and briefly introduce their security weaknesses. They are as follow:

A. ANALYSIS OF EXISTING SIP AUTHENTICAIONS:

S/MIME in SIP Authentication: SIP messages are capable of carrying the Multipurpose Internet Mail Extensions (MIME) bodies. Secure/Multipurpose Internet Mail Extensions (S/MIME) is used for providing end-to-end confidentiality and integrity of MIME contents to some extend by replicating header fields in MIME part [13]. However, replication of all header fields inside MIME part exposes some problems. First SIP header field might get altered by intermediate SIP entities which make it difficult for recipients to identify legal or malicious changes in header. Second, SIP messages can be large by their size which causes overhead for processing and transporting of messages [13].

Certificate-based less mutual Authentication:it is an approach that eliminates the need of PKI even though it is important for users to exchange their public keys. It will make the use of a PKG, but the PKG only generates a component of user private key, while second component is generated by user. The drawback of this authentication is that it has no process of revoking keys [13].

SIP Digest Authentication: As we mentioned in section 2, SIP Digest is known as challenge-response which is used as authentication for client to client or client to proxy [13]. Here the recipients can challenge the sender identity using nonce value. To respond to this challenge, the sender sends a message digest calculated using nonce, username, user password, realm and other optional parameters. MD5 algorithm is used to compute response value unless another algorithm is specified in the realm parameter during challenge.In case the recipient is a proxy server, then 407 messages with Proxy-Authenticate header is send to challenge the client. Client includes a Proxy-Authentication in header to reply to that challenge. When receiving re-issued request, recipient verify that the authenticated user is authorized to perform action. Otherwise, authorization is failed. MD 5 is widely used cryptographic hash function. It has been employed in a wide variety of security applications and commonly used to check data integrity [14]. As a result we used this authentication mechanism combined with user identity and password to secure user session initiation in wire and wireless environment. With these combinations we provide acceptable security mechanisms which will provides authentication, information integrity, and confidentiality.

B. ANALYSIS OF DELAY

In this section we will analyze the delay which result from the number of messages sent (INVITE) by the UAC and received by UAS or Proxy Server as of (200 OK, ACK, or Media RTP). The total session setup delay for SIP over UDP is the time needed for all messages involved in the various transactions to be successfully received by UAC and UAS [16]. In other words, the session setup consists of the successful completion of the client-side and server-side transaction. Therefore, the total transmission delay for setting up the session (e.g. UAC1 and UAC2) is the addition of the delays for transmitting all N messages necessary to setup session using SIP over UDP [16].

$$Tt_{UDP} = \sum_{i=1}^{N} Tt(i)_{UDP} \tag{1}$$

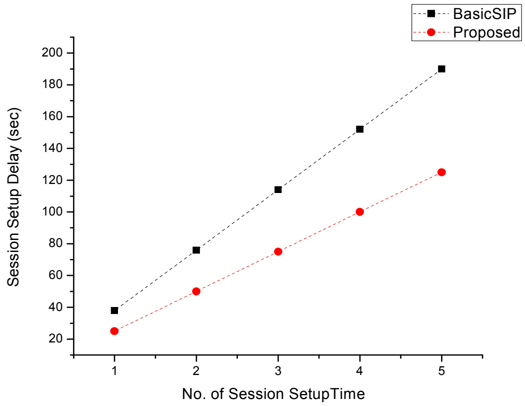
In analyzing delays, we are considering the following based in our proposed scenarios: 1) average Session Setup delay from the time UAC1 sent (INVITE) until UAS receive the requested content (200 OK, ACK). 2) User Authentication Delay vs. No of Session Setup Times between UAC1 and SIPPS. We will compare our approach with the basic Session Initiation [17,8]. Note when message is transmitted, the value will double after first sent INVITE messages have been sent. (For example the message sent from UAC1 to Proxy Server = 2msg multiply be time it took to receive them that will be 1second). Note that we also do not count the Media RTP as a message in our calculations. We assume that the no of messages between UAC1 and SIPPS is equal to 5 and the delay time is 1sec. So when you resend this message to other server, the delay will be doubled (e.g. 2second and so on). We also assume that the on. of messages exchanges between SIPPS and Cache is 5 for us, 10 for others and delay time in that period will be 4sec. Based in these values we calculate average delay for session setup (table. 3). We also assume that user might terminate session and initiate a new one;therefore we might have more than one session.

In Figure 7 we assume that the user eventually will terminate the session so we calculate the time that it will take the user to re-set up a session starting from the UAC sending INVITE message to the time UAS receive the 200 OK or ACK messages. As it is illustrated in the graph, our proposed mechanism present less delay comparing to Basic SIP which mean that we had a few messages exchange and that will optimize the use of resources. Furthermore, improve the session initiation and save the time needed to initiate the session.

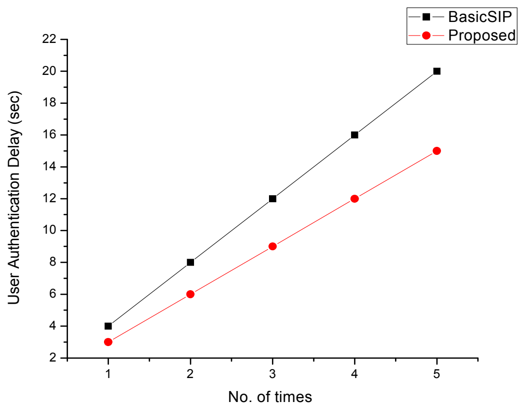
In figure 8, we calculate the accumulated user authentication delays against the number of times user initiation session setup (transfersession to another device). Our graph illustrates less delay when user authenticated to services before initiation of the session. SIP authentication security is based on the challenge-response mechanism. As result minimize the time needed and enhancing the performance.

(Table 3) Average Session Setup Delays

Delay time	Basic SIP	proposed
UAC1 to SIPPS	8ms	5ms
SIPPS to Cache	30ms	20ms
Average	38ms	25ms



(Figure 7) Accumulated Session Setup Delays vs. No of Session Setup Times.



(Figure 8) Accumulated User Authentication Delay vs. No of Session Setup Times.

First Message: UAC to SIP Proxy Server
INVITE sip:SCA@icns.com SIP/2.0
Via: SIP/2.0/UDP SIPPS@icns.com;
branch = j2cG3anachds1
Max-Forwards: 70
To: SCA sip: SCA@icns.com
From: UAC1 sip:UAC1@icns.com;
tag= 2927121663
Call-ID: b93c5d6dfdd621,
Cseq: 1INVITE
Content-Length:0

Second Message: SIP Proxy Server challenge UAC authentication
SIP/2.0 407 Proxy Authentication Requires
Via: SIP/2.0/UDP SIPPS@icns.com;
branch = j2cG3anachds1
To: SCA sip: SCA@icns.com
From: UAC1 sip:UAC1@icns.com;
tag= 2927121663
Call-ID: b93c5d6dfdd621, Cseq: 1INVITE
Proxy-Authenticate: Digest
Algorithm =MD5,
Realm="SIPPS@icns.com",
Nonce = "2c493d12bc4523",
opaque= "423bcb254c121d34cc"
Content-Length:0

Third Message: UAC acknowledges the challenge of 407 message and respond:
ACK sip: SCA@icns.com SIP/2.0
Via: SIP/2.0/UDP SIPPS@icns.com;
branch = j2cG3anachds1
To: SCA sip: SCA@icns.com
From: UAC1 sip:UAC1@icns.com;
Tag= 292712166, Call-ID: b93c5d6dfdd621,
Cseq: 1 ACK , Content-Length:0

Fourth Message: SIP Proxy Server to SCA
INVITE sip: SCA@icns.com SIP/2.0
Via: SIP/2.0/UDP SIPPS@icns.com;
Branch = j2cG3anachds1, Max-Forwards: 70
To: SCA sip: SCA@icns.com
From: UAC1 sip:UAC1@icns.com;
tag= 2927121663
Call-ID: b93c5d6dfdd621, Cseq: 2INVITE
Proxy-Authorization: Digest , Username="UAC1",
algorithm=MD5, Realm="SIPPS@icns.com",
Dnonce="1c473d122bc4523"
qop="423bcb254c121d34cc"
Response="948abc233cc3576792", nc=0000001,
SIPPSnonce ="4b48bcc",
Content-Length:0

(Figure 9) Description of message detail

C. ANALYSIS OF MESSAGE DETAILS OF PROPOSED SYSTEM ARCHITECTURE

In figure 9 we illustrate details of message exchange during SIP and their contents in sequence chart [8]. The first three messages present the authentication steps which occur between UAC and SIP Proxy Server in order to establish session initiation that allow user to receive playback of their contents. Furthermore, a verification of user id, password and realm will combined in this process to provide strong security. As a result preventing some of the security threats [9, 10,15]. In the fourth and the rest of the messages from SIP Proxy Server to Session Composition Agent we will use the same message steps to generate them as in the first message.

6. CONCLUSION

In this paper we have proposed a framework of N-Screen Session Manager based N-Screen Services using Cloud Computing for Thin-Client. We also design SIP UAC with the ability to allow user to have flexibility to playback contents (TV Drama, Ads, and Dialog etc) as well as allowing user to transfer their contents to variety of devices. N-Screen Session Manager is necessarily to manage session status of all communication sessions, manages handling of received requests and replies, allows a user to playback multimedia contents anytime with variety of devices for screen sharing, and allows a user to transfer an ongoing communication session from one device to another. We also discussed major security issues that occurs in Session Initiation Protocol as well as minimizing delay resulted from session initiation (playback or transfer session). As further research work, is to implement the mechanism in real time services. Providing more result to make sure that our mechanism cover more important aspect in our proposed scenarios.

ACKNOWLEDGEMENTS

"This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2011-0020515)"

REFERENCES

[1] J. Rosenburg, "SIP: Session Initiation Protocol," *IETF RFC 2543*, March 1999.

[2] J. H. Song, J. H. Byeon, N. M. Moon, "Design and Implementation of Metadata for n-Screen UCC Recommender System," *Multimedia and Ubiquitous Engineering (MUE), 2011 5th FTRA International Conference on*, vol., no., pp.263-266, 28-30 June 2011

[3] J. Nieh, S. J. Yang, N. Novik, "A Comparison of Thin-Client Computing Architecture," *Network Computing Laboratory, Columbia University*, Nov 2000.

[4] X. Wang, B. Wang, J. Huang,"Cloud computing and its key techniques," *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*, vol.2, no., pp.404-410, 10-12 June 2011

[5] M. Carroll, A. van der Merwe, P. Kotze, "Secure cloud computing: Benefits, risks and controls," *Information Security South Africa (ISSA), 2011*, vol., no., pp.1-9, 15-17 Aug. 2011

[6] S. Begum, M.K. Khan," Potential of cloud computing architecture," *Information and Communication Technologies (ICICT)*, *2011 International Conference on*, vol., no., pp.1-5, 23-24 July 2011

[7] S. Rajan, A. Jairath, "Cloud Computing: The Fifth Generation of Computing," *Communication Systems and Network Technologies (CSNT)*, *2011 International Conference on*, vol., no., pp.665-667, 3-5 June 2011

[8] Q. Qiu, "Study of Digest Authentication for Session Initiation Protocol (SIP)," *SITE*, *University of Ottawa*, *Ontario*, *Canada.* <http://www.site.uottawa.ca/~bob/gradstudents/DigestAuthenticationReport.pdf>

[9] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Jonston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, June 1999.

[10] S. Salsano, L. Veltri, D. Papalilo, "SIP Security Issues: The SIP Authentication Procedure and its Processing Load," Network IEEE, vol. 16, no. 6, pp. 38-44, Nov-Dec 2002. [Online] http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1081764&isnumber=23365

[11] A. Singh, A. Acharya," Multiplayer network gaming with the session initiation protocol", *Computer Networks, vol. 49, issue 1, 15 September, 2005, pg 38-51.*

[12] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawernce, P. Leach, A. Luotonen, L. Stewart, " HTTP Authentication: Basic and Digest Access Authentication," RFC 2617, June 1999.

[13] A. Alhasib, A. Azfar, Md. S. Morshed, "Towards Public Key Infrastructure less authentication in Session Initiation Protocol," *IJCSI, vol. 7, issue 1, no. 2, January 2010.*

[14] MD5 Message-Digest Algorithm, [Online] http://en.wikipedia.org/wiki/MD5

[15] P. Vesterinen, "User authentication in SIP," Helsinki University of Technology, [Online] http://www.tml.tkk.fi/Publications/C/22/papers/Vesterinen_final.pdf

[16] H. Fathi, S. S. Chakraborty, R. Prased, "Optimization of SIP Session Setup Delay for VOIP in 3G Wireless Networks," *IEEE Transaction on Mobile Computing, vol. 5, no. 9, September 2006.*

[17] K. H. Choi, W. M. Kim, J. K. Kim, K. S. Kim, "Design and Implementation of IMS Service Continuity between IPTV and Mobile, " *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, vol., no., pp.977-980, 13-16 Feb. 2011

◐ 저 자 소 개 ◑

아이만 압둘라 알사파르 (Aymen Abdullah Alsaffar)

2004년 Newbury College 졸업(공학사).
2011년 경희대학교 전자정보대학원 컴퓨터공학과 졸업(공석사).
2011년～현재 경희대학교 컴퓨터공학부 박사과정.
관심분야 : N-스크린, 클라우드 컴퓨팅, 가상 네트워크, IPTV, 씬클라이언트, 네트워크 보안.
E-mail : aymen@khu.ac.kr

송 표 (Song Biao)

2008년 Hebei Normal University (학사)
2010년～현재 경희대학교 대학원 컴퓨터공학과 석•박사 통합과정
관심분야 : task co-allocation, 클라우드 컴퓨텅과 개인 정보의 역동적인 협업.
E-mail : bsong@khu.ac.kr

모하마드 메히디 핫싼 (Mohammad Mehedi Hassan);

2003년 Islamic University of Technology (학사).
2010년 경희대학교 대학원 컴퓨터공학관 석•박사 통합과정 졸업(공박사)
2011년 3월～2011년 10월 경희대학교 대학원 컴퓨터공학학과 연구 교수
2012년～현재 King Saud University 조교수
관심분야: 클라우드 컴퓨텅, 데이터 집약적인 컴퓨팅, 미디어 클라우드, 모바일 클라우드, 게임 이론, 다이나믹 VM 자원 할당, IPTV, 가상 네트워크,센서 네트워크 및 시스템 가임 / 게시.
E-mail: mmhassan@ksu.edu.sa

허 의 남(Eui-Nam Huh)

2002년 The Ohio University 전산학과 졸업(박사)
2002년～2003년 삼육대학교 컴퓨터공학과 조교수
2003년～2005년 서울여자대학교 컴퓨터공학과 조교수
2005년～2012년 경희대학교 컴퓨터공학과 부교수
2012년～현재 경희대학교 컴퓨터공학과 교수
관심분야 : 클라우드/그리드 컴퓨팅, 센서 네트워크, 네트워크 보안, 모바일 컴퓨팅, etc.
E-mail : johnhuh@khu.ac.k