A linear chained approach for service invocation in IP Multimedia Subsystem

Jianxin Liao [a,b], Qi Qi [a,b] , Zhaoyong Xun [c], Tonghong Li [d], Yufei Cao [b], Jingyu Wang [a,b]

[a] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, PR China*
[b] *EBUPT Information Technology Co., Ltd., Beijing 100191, PR China*
[c] *Sun Kaisens Technology Co., Ltd., Beijing 100071, PR China*
[d] *Technical University of Madrid, Madrid 28660, Spain*

ABSTRACT

IP Multimedia Subsystem (IMS) is considered to provide multimedia services to users through an IP-based control plane. The current IMS service invocation mechanism, however, requires the Serving-Call Session Control Function (S-CSCF) invokes each Application Server (AS) sequentially to perform service subscription profile, which results in the heavy load of the S-CSCF and the long session set-up delay. To solve this issue, this paper proposes a linear chained service invocation mechanism to invoke each AS consecutively. By checking all the initial Filter Criteria (iFC) one-time and adding the addresses of all involved ASs to the "Route" header, this new approach enables multiple services to be invoked as a linear chain during a session. We model the service invocation mechanisms through Jackson networks, which are validated through simulations. The analytic results verify that the linear chained service invocation mechanism can effectively reduce session set-up delay of the service layer and decrease the load level of the S-CSCF.

1. Introduction

IP Multimedia Subsystem (IMS) is introduced in The 3rd Generation Partnership Project (3GPP) release 5, which provides the users an abundance of voice, video, messaging and data services, such as multimedia call, video conference, network games and so on [1–3]. Currently, IMS is updated in release 11, and is considered to provide the same set of services to user terminals regardless of the use of fixed or mobile access technologies for the future networks. The service provision architecture in IMS supports the invocation of multiple services during a session according to the user's subscription. This architecture is efficient and flexible for the introduction of new services. The service provider can easily implement the new service by deploying an Application Server (AS) to IMS.

In this paper, the service invocation in IMS service provision architecture means the continuous procedure in which different services are invoked during a session. For the implementation of service invocation in IMS, the Serving-Call Session Control Function (S-CSCF) handles the execution of services and forwards the Session Initiation Protocol (SIP) request to each AS one by one. For example, suppose that Bob subscribes two services, Call Forwarding (CF) and Customized Alerting Tone (CAT) [4]. If Alice wants to set up a call with Bob, the 'INVITE' request is forwarded to Bob's home S-CSCF via the Interrogating-Call Session Control Function (I-CSCF). Then Bob's home S-CSCF forwards the request to the AS hosting CAT service. As the execution of CAT service logic is done, the AS returns the request to the S-CSCF. But Bob wants to forward any call to his friend Tom, then the S-CSCF forwards the request again to the AS hosting CF service. Thus in this case, the call set-up procedure between Alice and Bob contains two independent AS invoking flows, from the S-CSCF to CAT and from the S-CSCF to CF. Obviously, when the subscriber's service profile consists of several different applications, the current IMS service invocation mechanism is unable to invoke multiple services for only once during a session, causing excessive signaling traffic along with the long delay time.

Currently, the research for satisfying the requirement of decreasing the session set-up delay and lowering the load of network entities in IMS are mainly at two aspects: (1) increasing the capability of core network entities; (2) modifying basic session set-up signaling flows [5–7]. Yet there is little work about improving service invocation mechanism in the service layer to optimize IMS performance. The definition of the Service Capability Interaction Manager (SCIM) is introduced in 3GPP specification release 5. It is a specialized type of SIP AS, which lessens the burden of the S-CSCF by performing the role of interaction management of ASs [3]. However, from 3GPP release 5 to 3GPP release 11, the specifications do not explain the architecture and functionalities of SCIM. In the meantime, the service invocation and interaction management provided by SCIM are not standardized.

In this paper, we propose a novel service invocation mechanism based on the linear chained service invoking process, to optimize the IMS service provision when multiple services need to be invoked during a session. The proposed mechanism supports the AS to AS direct service invocation. Comparing with the existing approaches, our mechanism has the following four advantages. Firstly, it reduces the session set-up delay, as the signaling between the S-CSCF and each AS during the service invocation is avoided. Secondly, the load level of the S-CSCF is decreased, especially when excessive services are involved in a session set-up procedure. Thirdly, it enables the S-CSCF to invoke multiple services only once during a session, and avoids the cost of signaling interaction between SCIM and ASs for the Distributed SCIM (DSCIM) service invocation mechanism in [8]. And finally, the new mechanism only requires the S-CSCF to insert the addresses of ASs to the header of SIP request without any modification of the existing IMS core network.

The remainder of this paper is structured as follows. Section 2 surveys the related work. After describing the current IMS service invocation mechanism in Section 3, we present the linear chained service invocation mechanism in detail in Section 4. Section 5 models the service invocation mechanisms by Jackson networks. Section 6 evaluates the performance of our proposed new mechanism in terms of the mean session set-up delay and the load of the S-CSCF. Finally, we conclude this paper and discuss our future work in Section 7.

2. Related work

In the literature, several mechanisms have been proposed for service provision in IMS. Pavlovski [9] introduces a service delivery platform in IMS for generating multimedia content to the users. Also, O'Connell [10] introduces a horizontal service delivery architecture for next generation IP multimedia applications in IMS. To realize the dynamic composition of distributed service entities in IMS, Lavinal et al. [11] propose a service overlay architecture in which a service level path is dynamically established to fulfill the user's requirements. Lee [12] introduces a method, which allows third-party application developers to quickly create new applications by using IMS building blocks and Web 2.0 technologies. Munasinghe et al. [13] propose a mobility-aware interworking architecture to ensure uninterrupted service continuity for handoff sessions in heterogeneous IMS environment. Moreover, some researches improve the performance of the provision of some services in IMS, such as the work in [14,15].

How to efficiently invoke services in a session is a key issue for service provision in IMS. Several service invocation mechanisms have been proposed to improve the IMS performance [16–21]. However, up to now, no IMS service invocation mechanism exists, which can reduce the session set-up delay while lowering the load of S-CSCF without new network entity involved.

Gouya et al. [16,17] propose a service invocation approach through SCIM, which adds the Service Capabilities (SCs) invoking flows to the current S-CSCF service invocation mechanism. The SCIM requests the subscriber's service profile from the S-CSCF and creates SC filter criteria based on the operator's policy. The service invoking procedure for an integration service is divided into two parts: (1) the S-CSCF invokes ASs in accordance with initial Filter Criteria (iFC); (2) the SCIM invokes SCs in accordance with SC filter criteria. However, this approach increases the session set-up delay significantly.

In our previous work [18], we have proposed a Call-state-based Application Triggering Architecture (CATA). The proposed call-state-based Filter Criteria (cFC) are stored in the Home Subscriber Server (HSS) as part of the user profile and can be downloaded to the S-CSCF. The CATA avoids the signaling traffic by checking the relative cFC in case that no service logic is triggered in the AS during the session set-up procedure. Also, Chiang et al. [19] propose a distributed service invocation function (DSIF) to eliminate the unnecessary relay from S-CSCF to AS. The DSIF works as an overlay on the top of the current IMS network, which relays the SIP signaling to the AS only when the service really needs to be executed. However, both CATA and DSIF can only reduce the signaling traffic when the service in an AS does not need to be executed, and thus cannot essentially solve the problem of long set-up delay and heavy load of the S-CSCF.

To reduce the session set-up delay in IMS service layer, we have proposed a distributed SCIM (DSCIM) [8] service invocation mechanism based on the recommendations of architecture and functionalities of SCIM [20]. The DSCIM enables multiple services to be invoked consecutively with no signaling being forwarded back to the S-CSCF. However, the SCIM Node (SCIMN) embedded in each AS increases the complexity of the network operation and also brings about the extra signaling cost.

To improve the system performance, we have proposed a Group based Service Triggering Algorithm (GSTA) [21], which introduces the conception of "AS Grouping" by dividing the services into groups. When a group is triggered, the services within the group are invoked consecutively. Thus, the GSTA can reduce the number of hops in the signaling path as well as the signaling traffic load of the S-CSCF. However, GSTA can only improve the performance of the service invocation when all the services in a session belong to a group.

3. Standard service invocation mechanism in IMS

The S-CSCF is the functional entity in the IMS core network for session control and service invocation. It can invoke multiple multimedia applications in a session and provide a user a coherent and consistent IP multimedia service experience.

3.1. The iFC model for service invocation

The S-CSCF applies the filter criteria to determine the need to forward SIP requests to ASs, which is downloaded from the HSS as part of the user profile. The iFC represents a provisioned subscription of a user to some services, which are valid throughout the user registration lifetime or until the user profile is changed. If there are multiple iFCs assigned for one subscriber, the *Priority Number* which is an attribute of iFC class describes the order in which the S-CSCF assesses them. The higher the *Priority Number* the lower the priority of the filter criteria is; i.e., the filter criteria with a higher value of *Priority Number* should be assessed after the filter criteria with a smaller *Priority Number* have been assessed. Moreover, each instance of iFC class is composed of zero or one instance of *Trigger Point* (TP) class along with one instance of *Application Server* class. The TP class describes the trigger points that should be checked in order to find out if the indicated ASs should be contacted or not. And the TP class is composed of one or more instance of *Service Point Trigger* (SPT) class. The *Application Server* class defines the AS, which is contacted if the TPs are met, and its attribute *Server Name* describes the SIP Uniform Resource Locator (URL) of the AS. Other attributes of iFC can be referenced in [3,22].

3.2. The service invocation procedure

When the S-CSCF receives SIP requests sent by the User Equipment (UE) from the I-CSCF, it identifies which applications are invoked per subscriber; understands the appropriate order of the set of applications; and resolves the service interactions during the session. The S-CSCF firstly downloads the relevant set of iFCs from the HSS, and then checks whether the SIP request matches the SPT instances of the iFC with the highest priority. If so, the S-CSCF forwards the request to the AS specified by this iFC. After the AS finishes its processing, it sends the request back to the S-CSCF. Afterwards, the S-CSCF checks the next following iFC with lower priority. The invocation process repeats until the multiple services in the user profile are all invoked. For example, in Fig. 1, the standard invocation procedure is S-CSCF → AS#1 (service A) → S-CSCF → AS#2 (service B) → S-CSCF → ... → S-CSCF → AS#*m* (service X) → S-CSCF.

3.3. Shortcomings in standard service invocation mechanism

To provide value-added services to the end user, IMS service provider may dynamically compose distributed service entities [11]. In this case, the number of invoked ASs during one session is always larger than one. On the other hand, some common service capabilities in the AS are abstracted as special re-usable elements called SCs, which can be used to create new services and be deployed independently in special AS [23]. Thus, some services are divided into several executions of SCs, which also results in multiple service invocation. In the above two scenarios, obliviously, more than one ASs needs to be sequentially contacted during an IMS session set-up procedure. Moreover, consider a particular case that an AS providing

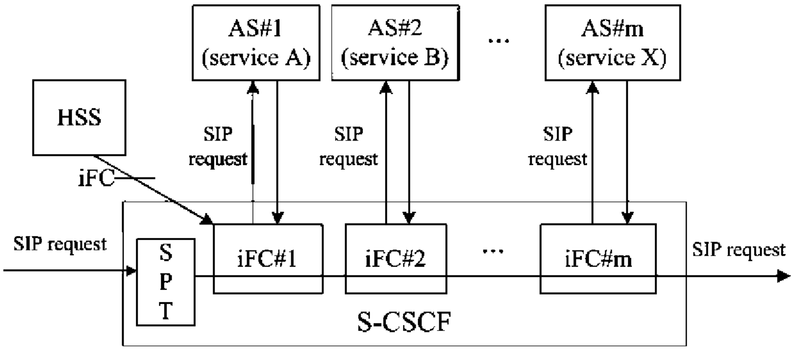

Fig. 1. Standard service invocation model.

more than one services, for example, AS#1 provides service A and B. If a session involves service A and B, in the standard service invocation the S-CSCF should interact with the same AS twice by instructing it to execute the corresponding service logic [2]. As the SIP request is sent back to the S-CSCF after each AS finishes its processing, the standard service invocation mechanism may lead to long session set-up delay as well as the heavy load of the S-CSCF.

4. Linear chained service invocation mechanism

4.1. The main idea

In case of multiple service invocation, assume that there are m subscribed services to be invoked in a session. The linear chained service invocation mechanism improves the standard flow, as depicted in Fig. 2. When the S-CSCF receives the initial SIP message, it evaluates the iFCs in terms of their priorities and determines the sequence of ASs to be contacted. Then, the request is transmitted to the AS with the first priority, i.e. AS#1, and after the execution of the corresponding service logic, the request is directly forwarded to the AS with the second priority, i.e. AS#2, instead of being forwarded back to the S-CSCF. When the AS with the last priority, i.e. AS#m finishes the execution of service logic, it finally forwards the request back to the S-CSCF. Thus, the optimized flows of invoking m services in a session is S-CSCF → AS#1 (service A) → AS#2 (service B) → ... → AS#m (service X) → S-CSCF. In the new service invocation mechanism, the signaling flow from the first invoked service to the last invoked one is like a linear chain.

However, the service invocation algorithm of 3GPP standard cannot support the optimized flow. As an AS cannot obtain the address of any other AS, the SIP request cannot be forwarded from the current AS to the next one without the involvement of the S-CSCF. Therefore, we consider to use the "Route" header of the SIP request to record the addresses of all the contacted ASs so that the S-CSCF can invokes multiple services at a time.

4.2. The multi-service identifier in "Route" header

The S-CSCF controls the service invocation through the SIP URL of the AS hosting the invoked service logic. For example, the *ServerName* attribute "service A@as1.home.net" indicates that the SIP request should be routed to the service A of AS#1. In our proposed linear chained service invocation mechanism, the "Route" header in SIP includes the information about the sequence of ASs to be contacted. The S-CSCF firstly checks whether the SIP request matches the SPT instances of the iFC with the highest priority. If so, the SIP URL of the AS specified by this iFC is appended to the "Route" header of the SIP request. The S-CSCF checks all iFC instances in terms of their priorities one by one and appends the SIP URLs of all contacted ASs to the "Route" header of the SIP request. Once this process is performed, the SIP request is forwarded to the AS with the first priority. After the specified service logic is executed, the SIP request is forwarded directly to the AS with next priority, based on the indication of the SIP URL in the "Route" header.

In addition, to determine whether the next service is located in the same AS as the current service, the AS compares the string after the character "@" in the domain name of SIP URL:

(1) If two different services are located in the same AS, the request is not forwarded to the S-CSCF, and the next service is performed in the current AS.
(2) If the next service is located in another AS, the request is forwarded to that AS under the indication of the SIP URL in the "Route" header.

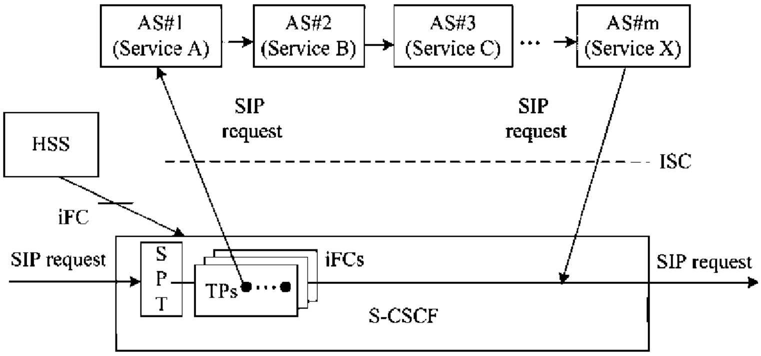

Fig. 2. Linear chained service invocation model.

4.3. The linear chained service invocation algorithm

The proposed service invocation algorithm runs in the S-CSCF and each AS. The algorithm in S-CSCF is depicted as follows:

```
S-CSCF receives SIP request message M from user U;
If M = REGISTER || M = session originating request && U is unregistered || M = session terminating request && U is
  unregistered
  S-CSCF download all the iFCs of user U from HSS;
End If
Sort out these iFCs according to their priorities;
Parse M;
Find out all the SPTs included in M;
For each iFC in the sorted iFCs
  If there is no TP instance in current iFC
    Insert AS ServerName to the "Route" header of M;
  Else
    Check the SPTs in TP and compare them with the SPTs in M;
    If match
      Insert AS ServerName to the "Route" header of M;
    End If
  End If
End For
Forward M to the first AS through ISC Interface;
```

The algorithm for the AS is depicted as follows:

```
AS receives M;
Get the first ServerName F in the "Route" header of M;
Execute the service logic of the service specified in F;
Get the next ServerName N in the "Route" header of M;
Remove the first ServerName from the "Route" header of M;
If the string after "@" in F equals to the string after "@" in N
  Forward M to itself;
If N == S-CSCF
  Forward M to the S-CSCF;
Else
  Forward M to the AS specified in N;
End If
```

When the S-CSCF receives a user's registration request or an originating request or a terminating request, it downloads the iFCs of that user from the HSS. The iFCs are ordered according to their priorities. Then, the S-CSCF checks all iFC instances one by one. If there is no TP instance in the iFC, or the TP in the iFC matches the SPT instances in the request, the *ServerName* of the iFC is added to the "Route" header of the request. Continually, the request is forwarded to these ASs one by one according to the indication of *ServerName* in the "Route" header.

4.4. Service interaction management and conflict resolution

When two or more services are invoked during a session, service interaction may happen. As the conflicts between invoked services may result in incorrect or unexpected behaviors, the service invocation in IMS must be supplied with a mechanism for the service interaction management.

The linear chained service invocation mechanism supports the service interaction management by detecting and dealing with the service conflicts. There is a service interaction database in S-CSCF, which stores a set of service conflicts and their resolutions [24]. When the S-CSCF adds the *ServerName* to the "Route" header, it inquires the service interaction database and checks whether or not the new service conflicts with the services in the "Route" header. In the case of confliction, the S-CSCF resolves it according to the suggestion defined in the service interaction database.

5. IMS service invocation model

The standard service invocation and linear chained service invocation mechanisms can be modeled as Jackson networks. The S-CSCF parses the SIP message, checks the service invocation conditions and obtains the addresses of matched ASs, which can be modeled as an M/M/1 queue. The execution of each service in an AS is regarded as an M/M/1 queue and modeled as independent node, even if two services are in the same AS [2]. For the simplification and without loss of generality, we assume:

(1) Each session only contains an initial request and its corresponding response. The session arrival rate is the initial SIP request arrival rate.
(2) All the users with the same registration profile belong to the same type, which have the same iFCs. Assume that there are T user types. For each user type t ($1 \leqslant t \leqslant T$), there are n_t iFCs, which means that at most n_t services can be invoked during a session. Also, for each user type t, n_t iFCs are ordered according to their priorities and numbered from 1 to n_t.
(3) There are m ($m \geqslant n_t$) services in the IMS network, numbered from 1 to m. We define a function $f(i,t)$ ($1 \leqslant i \leqslant n_t$, $1 \leqslant t \leqslant T$) to denote the identification number of the service specified by the i^{th} iFC of user type t. During a session, the next service to be invoked should have lower priority than the current one.
(4) The request arrives at the queues of Jackson network according to a Poisson process; and the service time in each queue within the Jackson network is exponentially distributed.

We define the parameters that are shown in Table 1.

5.1. The standard service invocation model

The standard service invocation mechanism can be modeled as a Jackson network including the S-CSCF and m services with feedback messages. The model for user type t is shown in Fig. 3. The S-CSCF processes two types of requests: (1) the SIP request received from the outside network. After receiving the new requests, the S-CSCF forwards it to the service#$f(i,t)$ with the probability q_i^t; (2) the SIP request sent back from service nodes. After finishing the execution of its service logic, the service node sends the SIP request back to the S-CSCF. When the S-CSCF receives the SIP request from the service#$f(i,t)$, it can forward the request to the service#$f(j,t)$ ($i < j \leqslant n_t$) with the probability $q_{i,j}^t$ or let the request leave the service layer with the probability $q_{i,0}^t$. As the iFCs are evaluated in terms of their priorities, the next service node to be contacted cannot have higher priority than the current one. Consequently, the total request arrival rate of the S-CSCF is the sum of the arrival rate of the requests from outside network and that of the feedback messages from each AS. Also, the request arrival rate of service#$f(i,t)$ is the sum of the arrival rate of the requests from outside network and that of the feedback messages from service#$f(j,t)$ ($1 \leqslant j < i$).

Then, we have:

$$\begin{cases} \lambda_1^t = q_1^t \lambda^t \\ \lambda_i^t = q_i^t \lambda^t + \sum\limits_{j=1}^{i-1} q_{j,i}^t \lambda_j^t \quad (1 < i \leqslant n_t) \\ \lambda_0^t = \lambda^t + \sum\limits_{i=1}^{n_t} \lambda_i^t \end{cases} \tag{1}$$

From (1), λ_1^t can be obtained directly. After λ_1^t is given, λ_2^t can then be deduced. λ_i^t can be deduced after $\lambda_j^t (1 \leqslant j \leqslant i-1)$ is given. Finally, λ_0^t can be deduced after λ_i^t ($1 \leqslant i \leqslant n_t$) is given.

The total request arrival rate of the S-CSCF is:

$$\lambda_0 = \sum_{t=1}^{T} \lambda_0^t \tag{2}$$

Table 1
Parameters of the Jackson network model.

Symbol	Definitions
λ	Initial request arrival rate of the S-CSCF from the I-CSCF
λ_0	Total request arrival rate of the S-CSCF
μ_0	The service rate of the S-CSCF
λ_k	Total request arrival rate of the service#k ($1 \leqslant k \leqslant m$)
λ^t	Initial request arrival rate of the S-CSCF from the I-CSCF for user type t
λ_0^t	Total request arrival rate of the S-CSCF for user type t
λ_i^t	Request arrival rate of the service specified by the i^{th} iFC for user type t, i.e. service#$f(i,t)$ ($1 \leqslant i \leqslant n_t$)
μ_k	The service rate of the service#k ($1 \leqslant k \leqslant m$)
q_i^t	The probability that the S-CSCF forwards request to the service specified by the i^{th} iFC for user type t, i.e. service#$f(i,t)$ ($1 \leqslant i \leqslant n_t$)
$q_{i,j}^t$	The probability that the service#$f(i,t)$ invokes the service#$f(j,t)$ ($i, j = 1, \ldots, n_t$; $i < j$)
$q_{i,0}^t$	The probability that the request leaves the network after service#$f(i,t)$ finishes the execution of its service logic
ρ_0	The load level of the S-CSCF
ρ_k	The load level of the service#k

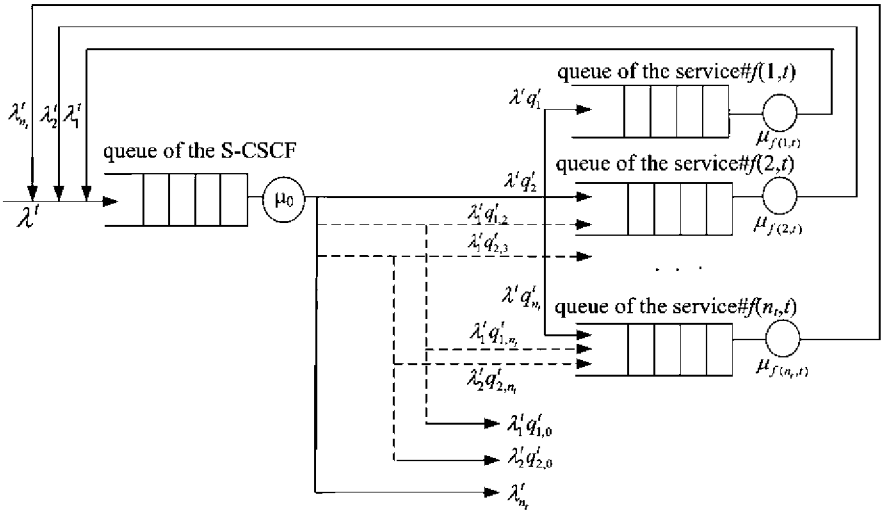

Fig. 3. Model of the standard service invocation mechanism.

And the total request arrival rate of service#*k* is:

$$\lambda_k = \sum_{t=1}^{T} \lambda^t_{i\,\text{with} f(i,t)=k} \tag{3}$$

When the S-CSCF and all services node with M/M/1 queue are stable, they constitute a Jackson network including (m + 1) nodes. According to the Jackson network theory, the mean delay of the standard IMS service invocation is:

$$T = \frac{1}{\lambda} \sum_{k=0}^{m} \frac{\rho_k}{1-\rho_k} \tag{4}$$

Here, $\lambda = \sum_{t=1}^{T} \lambda^t$ denotes the total arrival rate from outside network, i.e. the initial request arrival rate of the S-CSCF from the I-CSCF for all users, and $\rho_k = \lambda_k/\mu_k$.

5.2. The linear chained service invocation model

The model for user type *t* in the linear chained invocation mechanism is shown in Fig. 4. When the S-CSCF receives the SIP request from outside network, it forwards the request to service#*f*(*i*,*t*) with the probability q^t_i. After service#*f*(*i*,*t*) finishes the execution of its service logic, it forwards the request to service#*f*(*j*,*t*) ($i < j \leqslant n_t$) with the probability $q^t_{i,j}$ or sends the request back to the S-CSCF with the probability $q^t_{i,0}$. If the S-CSCF receives the SIP request from the service node, it lets the request

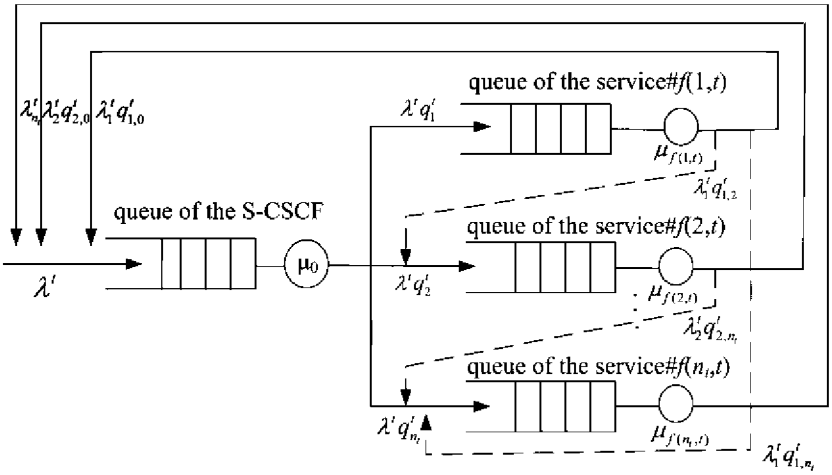

Fig. 4. Model of linear chained service invocation mechanism.

leave the service layer immediately. Thus, the request arrival rate of service#*f*(*i*,*t*) is the sum of the arrival rate of the requests from the S-CSCF and that of the feedback messages from AS#*f*(*j*,*t*) ($1 \leqslant j < i$).

Then, we have

$$\begin{cases} \lambda_1^t = q_1^t \lambda^t \\ \lambda_i^t = q_i^t \lambda^t + \sum\limits_{j=1}^{i-1} q_{j,i}^t \lambda_j^t \quad (1 < i \leqslant n_t) \\ \lambda_0^t = \lambda^t + \sum\limits_{i=1}^{n_t} q_{i,0}^t \lambda_i^t \end{cases} \tag{5}$$

From (5), λ_1^t can be obtained directly. After λ_1^t is given, λ_2^t can be deduced. And λ_i^t can be deduced after $\lambda_j^t (1 \leqslant j \leqslant i-1)$ is given. Finally, λ_0^t can be deduced after λ_i^t $(1 \leqslant i \leqslant n_t)$ is given.

After obtaining λ_i^t $(0 \leqslant i \leqslant n_t)$, λ_0 and λ_k $(1 \leqslant k \leqslant m)$ for the linear chained service invocation mechanism can be calculated by using Eqs. (2) and (3), respectively.

To compute the mean delay of the network for the linear chained service invocation, we can use Eq. (4) after obtaining λ_k $(0 \leqslant k \leqslant m)$.

5.3. Simulation validation

In order to validate the above analytic models, we have developed a Java-based test bed to simulate the IMS service invocation mechanisms, which includes several discrete-event simulators, SIP servers for the IMS core network and two ASs. The IMS network entities and ASs are all developed based on the open source code SIP stack SIPp [25]. We implement the CAT service in AS#1 and the CF service in AS#2. The iFCs are described as XML files, which are stored in HSS. The discrete-event simulator generates session requests according to the Poisson process with mean rate $\lambda = 40$ calls per second (cps); while the S-CSCF, AS#1 and AS#2 process the requests with the mean service time $u_0 = 100$, $u_1 = 43$ and $u_2 = 60$, respectively. All users have two iFCs, which are classified into two user types: (1) type 1, the iFC with CAT has higher priority than the one with CF; (2) type 2, the iFC with CF has higher priority than the one with CAT. For user type 1, the percentage of generated sessions that invoke CAT, CF, and both CAT and CF is 40%, 50% and 10%, respectively ($q_1^1 = q_2^1 = 0.5, q_{1,2}^1 = 0.5$). For user type 2, the percentage of generated sessions that invoke CF, CAT, and both CF and CAT is 40%, 50% and 10%, respectively ($q_1^2 = q_2^2 = 0.5, q_{1,2}^2 = 0.5$).

Here, we only record the session set-up delay in the service layer, i.e. the processing delay incurred by the S-CSCF and the invoked ASs, without considering the processing delay incurred by the other IMS entities as well as the wireless link delay. Figs. 5 and 6 depict the session set-up delays measured in the standard IMS service invocation and the linear chained service invocation, respectively. The AS-IS values of the session set-up delays and their mean values are also shown in the figures.

From the above analytic models, the session set-up delay under the above parameters for the standard IMS invocation and the linear chained invocation are calculated as 0.172 and 0.14, respectively. Both of them are approximately equal to the mean values of 1500 samples in Figs. 5 and 6, respectively, which demonstrate that our analytic model and simulation results are consistent in terms of the mean session set-up delay. Also, we can see from Figs. 5 and 6 that the delay for the linear chained service invocation is smaller than that of the standard mechanism. In addition, the variation of the session set-up delay for the linear chained service invocation is much smoother than that for the standard service invocation.

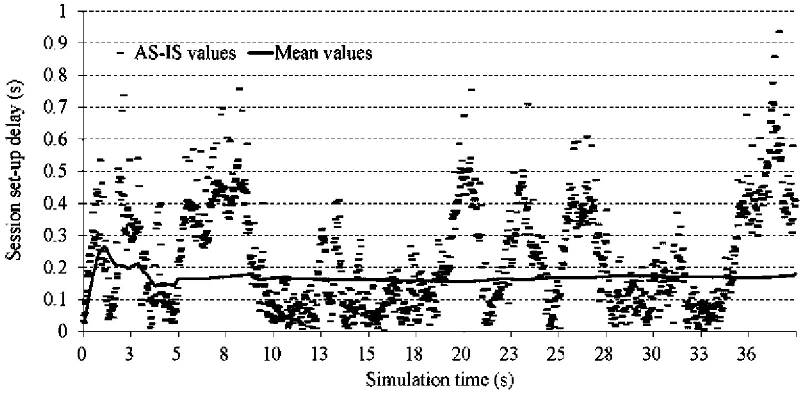

Fig. 5. Delay for the standard service invocation with $\lambda = 40$.

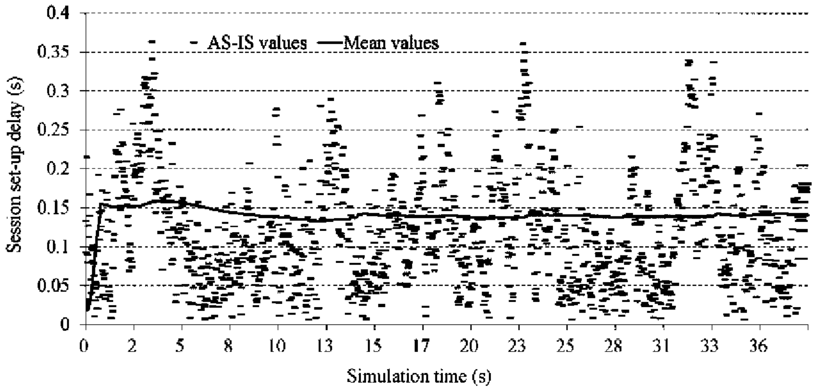

Fig. 6. Delay for the linear chained service invocation with $\lambda = 40$.

6. Performance analysis

In this section, we use the numerical examples to compare the performance of the linear chained service invocation with the standard service invocation in terms of: (1) session set-up delay; (2) S-CSCF load. In addition, we compare the cost of the linear chained service invocation and the DSCIM service invocation by simulation.

We set the service rate for the S-CSCF (μ_0) and each service to be 120 and 100 cps, respectively. We consider the following two scenarios: (1) scenario 1: there are two services ($m = 2$). There is only one user type with two iFCs ($T = 1$, $n_1 = 2$), where the first iFC contains service#1 and the second iFC contains service#2. The probabilities that the S-CSCF forwards the request to service#1 and service#2 are both 1/2 ($q_1^1 = q_2^1 = 1/2$), and the probability that service#1 invokes service#2 is 1/2 ($q_{1,2}^1 = 1/2$); (2) scenario 2: there are three services ($m = 3$). There are two user types ($T = 2$) with the same initial session arrival rate ($\lambda^1 = \lambda^2$). For user type 1, the user has 3 iFCs such that the first iFC contains service#1, the second iFC contains service#2 and the third iFC contains service#3. For user type 2, the user has 3 iFCs such that the first iFC contains service#3, the second iFC contains service#1 and the third iFC contains service#2. For user type 1, the probability that the S-CSCF forwards the request to each service is 1/3 ($q_1^1 = q_2^1 = q_3^1 = 1/3$); the probability that service#1 invokes service#2 and service#3 are both 1/3 ($q_{1,2}^1 = q_{1,3}^1 = 1/3$); and the probability that service#2 invokes service#3 is 1/2 ($q_{2,3}^1 = 1/2$). For user type 2, the probability that the S-CSCF forwards the request to each service is 1/3 ($q_1^2 = q_2^2 = q_3^2 = 1/3$); the probability that service#3 invokes service#1 and service#2 are both 1/3 ($q_{1,2}^2 = q_{1,3}^2 = 1/3$); and the probability that service#1 invokes service#2 is 1/2 ($q_{2,3}^2 = 1/2$).

6.1. Session set-up delay

Fig. 7 shows the mean session set-up delay under different session arrival rates for scenario 1 and 2. We can see that the delays for the linear chained service invocation are less than those for the standard service invocation. Along with the

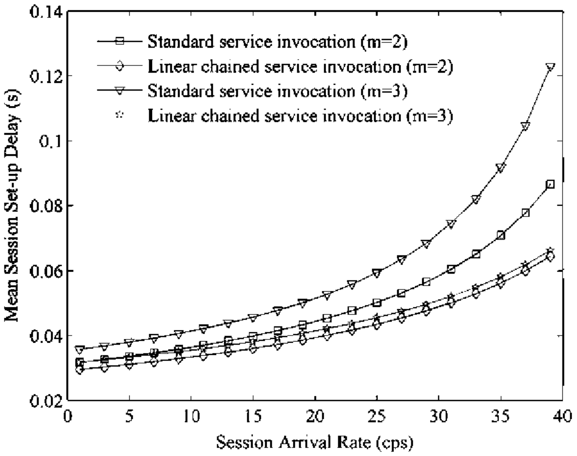

Fig. 7. Mean session set-up delay vs. when $m = 2$ and $m = 3$.

increase of the session arrival rate, the advantage of the linear chained service invocation becomes more significant. The reason is as follows: in the standard service invocation, the increase of the session arrival rate from outside network results in more feedback messages sent to the S-CSCF, which causes the S-CSCF to become a bottleneck. However, in the linear chained service invocation, with the increase of the session arrival rate from outside network, each service node's arrival rate increases equally, and thus the session set-up delay increases smoothly.

Fig. 8 shows the impact of $q^1_{1,2}$ on the mean session set-up delay under scenario 1. We can see that the mean session set-up delay of the standard service invocation increases significantly with the increase of $q^1_{1,2}$, while the mean session set-up delay of the linear chained service invocation increases slowly. The reason is as follows: in the standard service invocation, the requests should be firstly sent back to S-CSCF after executing the service logic of service#1, which results in the heavy load of the S-CSCF and the long session set-up delay. On the contrary, in the linear chained service invocation, service#1 forwards the request directly to serive#2, and thus its session set-up delay increases very slowly.

In order to study the impact of the number of invoked services on the mean session set-up delay, we figure out the following scenario 3: there is only one user type, which has n iFCs such that the ith iFC contains service#i. The probability that the S-CSCF forwards the request to service#i (q_i^1) is $1/n_t$. And the probability that service#i invokes service#j ($i < j$) is $1/(n-i + 1)$. Fig. 9 shows the change of mean session set-up delays by varying n from 2 to 20. We can see that the mean session set-up delay increases when the number of invoked service in a session increases. Furthermore, the mean session set-up delay for the standard service invocation grows exponentially, while the delay for the linear chained service invocation grows very slowly. Therefore, we can see the linear chained service invocation outperforms the standard service invocation when network operators want to provide users with composite services in which multiple services need to be invoked during a session.

6.2. S-CSCF load

The S-CSCF is an important entity in the IMS core network, and its load is one of key performance parameters for the network operators. Let ρ_0 denote the load level of the S-CSCF and p_c be the probability that the arrival request is rejected by the S-CSCF. Assume that the length of its queue is k. We have: $p_c = (1 - \rho_0)\rho_0^k$.

Fig. 10 shows the load level of the S-CSCF under different session arrival rates for scenario 1 and 2. Note that the service rate for the S-CSCF (μ_0) is set to be 90 cps for the purpose of overloading the S-CSCF. As the session arrival rate increases or the number of services grows, the load of the S-CSCF is aggravated. When $\rho_0 = 1$, the utilization ratio of the S-CSCF is 100%, which is denoted by the broken line in Fig. 10. For the standard service invocation, the S-CSCF becomes overloaded in scenario 1 when the session arrival rate is larger than 40, while the S-CSCF becomes overloaded in scenario 2 when the session arrival rate is larger than 36. With the increase of the number of services invoked in a session, the throughput of the system declines. On the contrary, for the linear chained service invocation, the number of services invoked during a session does not impact the load level of the S-CSCF, which is only determined by the session arrival rate.

Fig. 11 shows the change of p_c under different session arrival rates when k = 5 for scenario 1. We can see that the p_c for the linear chained service invocation is always lower than that of the standard service invocation, especially when the session arrival rate is more than 10 cps. In addition, Fig. 12 shows the values of p_c when k = 20 for scenario 3. We can see that the value of p_c for the linear chained service invocation does not increase as the number of iFCs increases.

6.3. Service invocation cost

Here, we make a comparison between the linear chained service invocation and the DSCIM [8] in terms of the signaling cost. Let C_{chain} and C_{DSCIM} be the service invocation cost associated with the linear chained service invocation and the DSCIM, respectively.

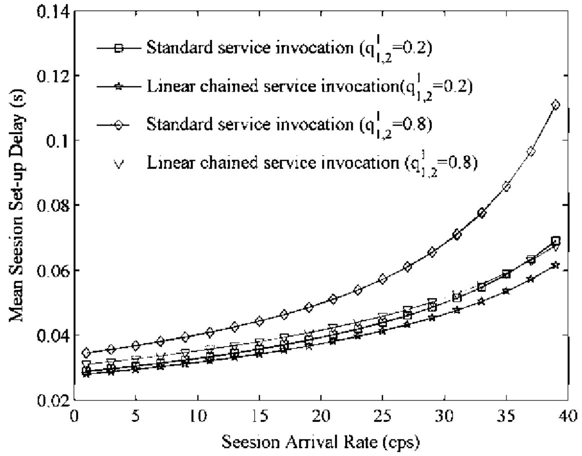

Fig. 8. Mean session set-up delay vs. λ when $q^1_{1,2} = 0.2$ and $q^1_{1,2} = 0.8$.

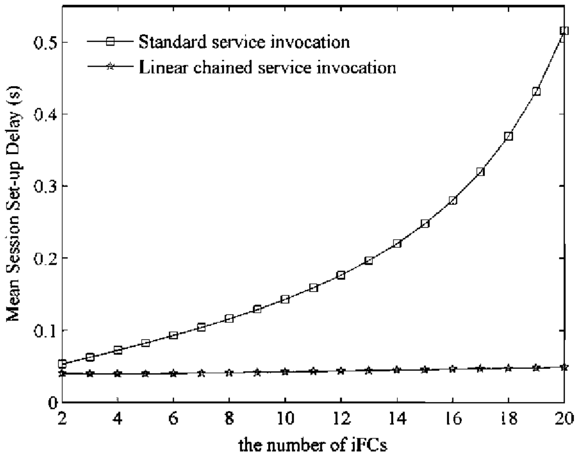

Fig. 9. Mean session set-up delay vs. *n*.

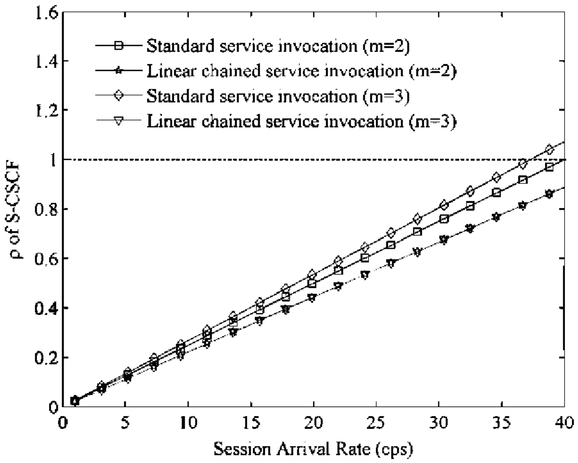

Fig. 10. Load of the S-CSCF.

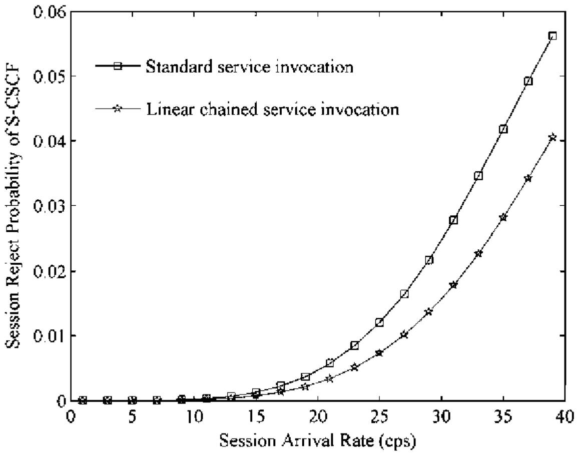

Fig. 11. The rejected probability of the S-CSCF vs. λ.

C_{DSCIM} can be essentially divided into three parts: (1) $C_{S\text{-}CSCF}$: the cost of the S-CSCF; (2) C_{AS}: the cost of the signaling transmission between ASs; (3) C_{SCIM}: the cost of the signaling interaction between the AS and the CSCIM for obtaining the address

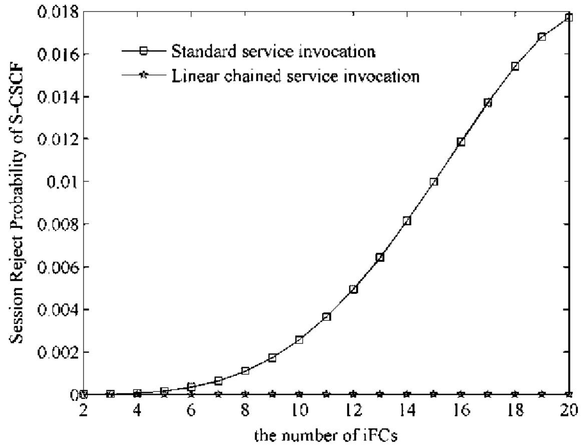

Fig. 12. The rejected probability of the S-CSCF vs. *n*.

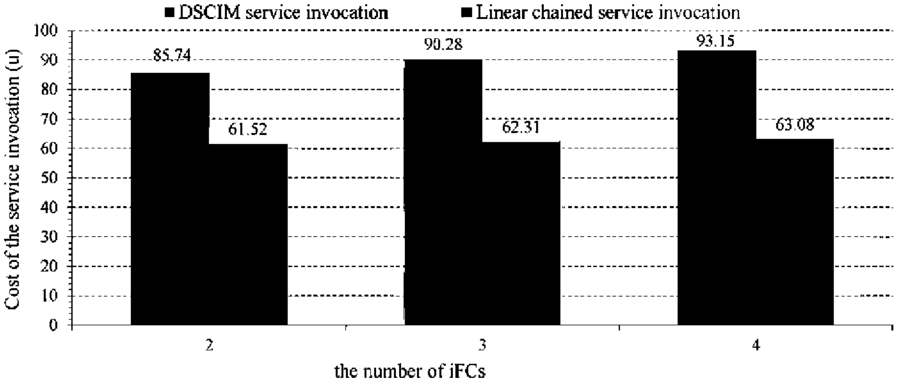

Fig. 13. The cost of service invocation.

of next AS. With respect to C_{chain}, the interaction between the AS and the CSCIM for obtaining the address of next AS is not required, due to the multi-service identifier in the "Route" header of SIP request. Therefore, the C_{chain} contains two parts: $C_{S\text{-}CSCF}$ and C_{AS}.

As the cost involving transmission expenses, bandwidth resources and other aspects, is difficult to measure, we assume that the cost unit is *u*. $C_{S\text{-}CSCF}$, C_{AS} and C_{SCIM} are set to be 30u, 5u and 20u, respectively. Fig. 13 shows the cost of the linear chained service invocation and the DSCIM under scenario 3. We can see that the linear chained service invocation significantly outperforms the DSCIM in terms of cost. The higher the number of services invoked during a session, the more advantage the linear chained service invoke brings about.

7. Conclusion

In this paper, we analyze how to efficiently invoke multiple services in a session for IMS. We have studied the standard mechanism and current improved proposals for IMS service invocation. In the standard IMS service invocation, the S-CSCF handles the execution of services and forwards the SIP request to each AS one by one, which causes excessive signaling traffic along with the long delay time. The improved proposals reduce the session set-up delay, but increase the load of the S-CSCF or introduce some new network entities. To solve these issues, we propose a linear chained approach which enables multiple services to be invoked consecutively by avoiding the signaling transmission between the S-CSCF and each AS. We present the design of the service invocation architecture, the multi-service identifier in the "Route" header and the detailed linear chained algorithm. We model the standard service invocation and our proposed approach through Jackson network and validate the analytic model by simulations. The numerical results under different network conditions show that the linear chained service invocation mechanism can effectively reduce the session set-up delay and lessen the load of the S-CSCF comparing with the standard service invocation mechanism. Also, the new approach decreases the service invocation cost comparing with our previous work. Moreover, the implementation of the new approach does not need any modification of IMS core network entities and signaling flows.

In this paper, we only propose the elementary method of the service interaction management and conflict resolution for the linear chained service invocation mechanism. Our future work is to investigate the detailed online algorithm for dealing with service conflicts, for example, extending the SIP header to record the rules created by the current invoked service so that the following invoked services can use them to detect the possible service conflicts [24].

References

[1] TS 23.002 V. 11.0.0. Network architecture, 3GPP, Sept. 2011.

[2] TS 23.228 V. 11.2.0. IP Multimedia System (IMS), 3GPP, Sept. 2011.

[3] TS 23.218, V11.0.0. IP Multimedia (IM) session handling; IM call model, 3GPP, Sept. 2011.

[4] TS 22.182, V11.0.0. Customized Alerting Tones (CAT) requirements; Stage 1, 3GPP, Mar. 2011.

[5] Umschaden K, Miladinovic I, Bessler S, Gojmerac I. Performance optimizations in UMTS packet switched call control. In: Proceedings of fifth IEE international conference on 3G mobile communication technologies. London: IEEE; 2004. p. 173–77.

[6] Foster G, Pous MI, Pesch D, Sesmun A, Kenneally V. Performance estimation of efficient UMTS packet voice call control. In: Proceedings of IEEE fall vehicular technology conference. Vancouver: IEEE; 2002. p. 1447–51.

[7] Cao Y, Liao J, Qi Q, Zhu X. A cache based session setup mechanism for IMS. In: Proceedings of IEEE ICC workshops. Beijing: IEEE; 2008. p. 261–65.

[8] Qi Q, Liao J, Zhu X, Cao Y. DSCIM: a novel service invocation mechanism in IMS. In: Proceedings of IEEE GLOBECOM. New Orleans: IEEE; 2008. p. 1–5.

[9] Pavlovski CJ. Service delivery platforms in practice. IEEE Commun Mag 2007;3:114–21.

[10] O'Connell J. Service delivery within an IMS environment. IEEE Veh Technol Mag 2007;1:12–9.

[11] Lavinal E, Simoni N, Song M. A next-generation service overlay architecture. Ann Telecommun 2009;64:175–85.

[12] Lee AY. Application creation for IMS systems through macro-enablers and web 2.0 technologies. Bell Labs Tech J 2010;15:23–52.

[13] Munasinghe KumuduS, Jamalipour Abbas. An analytical evaluation of mobility management in integrated WLAN-UMTS networks. Comput Electr Eng 2010;36:735–51.

[14] Buono Ao, Loreto S, Miniero L, Romano SP. A distributed IMS enabled conferencing architecture on top of a standard centralized conferencing framework. IEEE Commun Mag 2007;3:152–9.

[15] Vidal I, Garcia-Reinoso J, Soto I, Valera F. Evaluating extensions to IMS session setup for multicast-based many-to-many services. Comput Netw 2011;3:600–21.

[16] Gouya A, Grespi N. SCIM (service capability interaction manager) implementation issues in IMS service architecture. In: Proceedings of IEEE ICC. Istanbul: IEEE; 2006. p. 1748–53.

[17] Gouya A, Grespi N, Oueslati L. Next generation network service architecture in the IP Multimedia Subsystem. In: Proceedings of second Asian internet engineering conference. Pathumthani: ACM; 2006. p. 48–60.

[18] Liao J, Xun Z, Wang C, Zhu X. A call-state-based service triggering algorithm for IMS network. Int J Commun Sys 2009;3:343–64.

[19] Chiang W-K, Chang K-S. Design and implementation of a distributed service invocation function for the IP Multimedia Subsystem. Comput Commun 2011;9:1112–24.

[20] TR 23.810, V8.0.0. Study on architecture impacts of service brokering, 3GPP, Sept. 2008.

[21] Xun Z, Liao J, Zhu X, Cao CY. A group based service triggering algorithm for IMS network. In: Proceedings of IEEE ICC. Dresden: IEEE; 2009. p. 1–6.

[22] TS 29.228, V11.0.0. IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signaling flows and message contents, 3GPP, Jun. 2011.

[23] TR 181 004 V1.1.1. Telecommunications and internet converged services and protocols for advanced networking (TISPAN); NGN generic capabilities and their use to develop services, ETSI, Mar. 2006.

[24] Gouya A, Crespi N. Service invocation issues within the IP Multimedia Subsystem. In: Proceedings of IEEE ICNS 2007. Athens: IEEE; 2007. p. 33–8.

[25] SIPp [OL], <http://sipp.sourceforge.net/> [Oct. 2011].