

Performance Analysis of Recharging Scheme of M-SaaS through M-banking

¹Joseph Cosmas Mushi, ²Guan-zheng Tan, ³Felix Musau, ⁴Cheruiyot Wilson

^{*1, First Author}, Central South University, mushyjc@ieee.org

^{2, Second Author}, Central South University, tgz@mail.csu.edu.cn

^{3, Third Author}, Trusted Computing institute, CSU, musaunf@gmail.com

^{4, Third Author}, Central South University, wilchery68@gmail.com

doi : 10.4156/jcit.vol6.issue7.18

Abstract

Mobile SaaS (M-SaaS) is SaaS delivery model designed to use mobile network infrastructure for reserving and recharging credits of SaaS usage in pay-as-you-go manner for areas with poor financial services. The model integrates M-banking monetary feature into delivery process of SaaS so that potential customers in areas with poor access to financial services, such as online banking and credit cards, participate in SaaS delivery in real-time payment manner. In this paper, we present numerical analysis of M-SaaS recharging process, which together with M-SaaS credit-reservation scheme make M-SaaS model. The analysis shows that utilization and throughput of components involved in the scheme are healthy to avoid interruption of their involvement in other mobile network activities. However, load distribution at equilibrium leads to great imbalance, which needs further study for usefulness of M-SaaS model to reveal.

Keywords: Mobile SaaS, M-banking,, Recharging Scheme,
Performance Measures, Numerical Analysis

1. Introduction

Software as a Service (SaaS) is an emerging software distribution model that facilitates online delivery of application in utility manner and charged in pay-as-you-use [1,2]. In most cases the pay-as-you-use phenomenon entails real-time charging, which requires supportive financial services such as online banking and/or credit cards [3]. However, financial services in developing economy can hardly support online real-time payment [4], a factor that hinder implementation of SaaS thereto.

Mobile SaaS (M-SaaS) is a proposed model designed to use mobile network operators (MNOs) facilities to reserve and recharge credits of SaaS application usage in pay-as-you-use manner to those areas with poor access to financial services [5]. The model, which is made up of reservation and recharging schemes, integrates features of mobile-phone monetary service (M-banking) into SaaS distribution process so that potential customers who lives in areas with poor financial services can exploit SaaS applications.

In this work we present numerical analysis of M-SaaS recharging scheme, elaborated in [6], to determine usefulness of performance measures of involved components. The rest of the paper organized such that section 2 gives details of M-SaaS recharging scheme and criteria of its usefulness. The design of numerical analysis to determine performance measures of the scheme is given in section 3. Section 4 provides numerical examples, results and descriptive analysis of performance measures, before concludes in section 5.

2. M-SaaS Recharging Scheme

M-SaaS recharging scheme aim at automating recharging process of customer's pre-paid account through M-banking service when its balance falls below threshold. The scheme, together with M-SaaS reservation scheme, composed M-SaaS model that integrates mobile-phone network infrastructure into SaaS distribution process to solve challenge faces prospective SaaS customers in areas with poor financial service [5]

2.1. Traffic Pattern of M-SaaS Recharging Scheme

The dissemination in [5, 6] elaborate traffic pattern of M-SaaS recharging scheme, which referring to necessary routes taken by recharging request after generated by Account Balance Management Function (ABMF). The generation of recharging request is triggered by Session Based Charging Function (SBCF) whenever it receives feedback from ABMF and rate function (RF) that implicate credit insufficiency below threshold [6]. Figure 1, accompanied with stepwise description of sequence flow, gives an insight of recharging requests traffic pattern.

- Step 1.** SBCF generates trigger for recharging process after learn, in step 0, a credit-insufficiency reported from response of ABMF and RF.
- Step 2.** SBCF sends recharging trigger (recharge_trigger) to ABMF to instruct initiation of recharging process.
- Step 3.** ABMF generates recharging request (recharge_request) to initiate recharging process.
- Step 4.** recharge_request sent to recharging server (RS) for further top-up process.

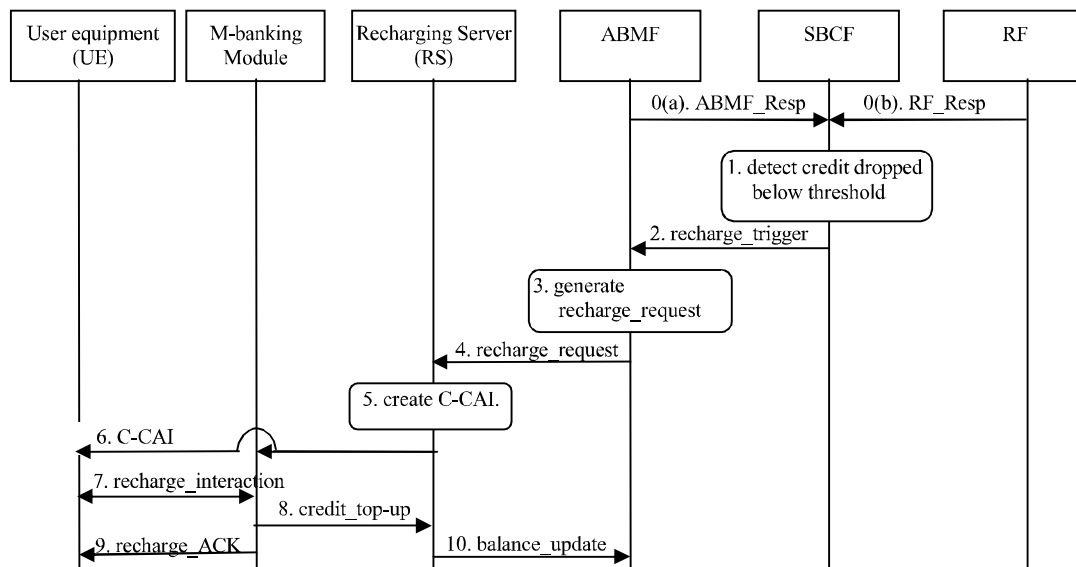


Figure 1. Sequence flow of M-SaaS recharging scheme

- Step 5.** RS creates customized charging advice information (C-CAI) to advise both M-banking and UE about a need to recharge credits in user pre-paid account.
- Step 6.** RS sends different version of C-CAI to M-banking and UE to prompt for further recharging process
- Step 7.** The recharging interaction between UE and M-banking involve following exchange:
 - a. UE initiate top-up process by sending credit top-up request to M-banking. User can ignore the prompt if does not wish to proceed to next SaaS session.
 - b. M-banking respond by querying back the top-up amount.
 - c. UE clarify top-up amount.
 - d. M-banking prepare top-up note and send to UE for approval and verification.
 - e. UE approve the note and verify by providing valid PIN code.
- Step 8.** If verification succeeds, M-banking module sends credit top-up note to RS to recharge user pre-paid account.
- Step 9.** M-banking module sends recharge acknowledgement (recharge_ACK) to customer.
- Step 10.** Recharging server updates ABMF with new balance.

2.2. Criteria for usefulness of M-SaaS recharging scheme

The major goal of M-SaaS recharging scheme is to avoid force termination of SaaS session due to credit insufficiency. Let S_i , ΔC , and Δt_R denote current SaaS session, credit-reservation window, and recharging window respectively. With reference to figure 2, if S_i has to be extended to S_{i+1} then the system must reserve usage credit for S_{i+1} before end of S_i , a process that last for ΔC_1 . According to M-SaaS reservation process [5], it is during ΔC_1 when SBCF would detect credit insufficiency for extension to S_{i+2} . In order to suffice the scheme's goal, Δt_R should complete before commencement of ΔC_2 .

Since the duration of ΔC_1 is approximately equal to ΔC_2 , then the usefulness of M-SaaS recharging scheme revealed if and only if Δt_R is less than session window (ΔS). However, Δt_R is basically depends on response time of components involved in recharging scheme. Therefore, the usefulness can be elaborated in terms of session window and mean response time, such that the total mean response time of components involved in recharging scheme must be less than ΔS .

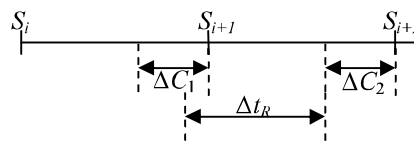


Figure 2. Graphical representation of windows in M-SaaS recharging scheme

In the next section, the study analyzes the performance metrics of components involved in M-SaaS recharging scheme, includes mean response time. The response time will suggest the choice of ΔS amid the challenge that shorter ΔS leads to fewer recharging traffic (due to less credits required for a session) but also squeezes Δt_R against longer total response time.

3. Analytical Model for Recharging Scheme

3.1. Design of Analytical Model

Consider a tagged recharging-request (denoted here by RR) that observed from the time it is generated until a successful top-up of credit. In M-SaaS model, such request is treated through multi-tier of steady-state functions triggered at SBCF after detected credits insufficiency. As shown in figure 3, recharging request propagates from ABMF through RS, M-banking and UE before sinks at ABMF.

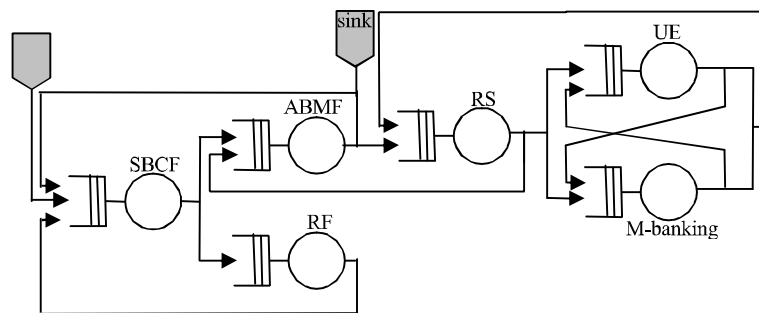


Figure 3. Network queuing model of M-SaaS recharging scheme

At each function, the request receives different treatments depend on state of the function and treatment that had received at its predecessor. The main modeling challenge herein stems from the fact that each request is treated in non-trivial terms based on instantaneous functional load. In order to face

the challenge we adopt Continuous-time Markov chain modeling technique based on steadiness behavior of these functions.

Accordingly, SBCF and RF are not much involved in core recharging process. For instance, RF is involved only to provide tariff-feedback to SBCF, which use the information to generate trigger. This means RF is involved even before generation of trigger, which implies its involvement does not affect the total duration of recharging process. On the other hand, the generation of trigger generally takes a short fraction of time, which consequently makes involvement of SBCF less significant in recharging window. Therefore, this analysis found useful to absorb less significant states for the purpose of reducing computation complexity and memory usage.

The fact that RR is usually generated and sink are clear evidence that M-SaaS recharging scheme involves both open and closed product-form queuing networks as elaborated in [7]. Figure 4 shows open and closed queuing networks of M-SaaS recharging scheme. Note that the figure involves only non-absorbed states based on reasons stated in previous paragraph.

Based on queuing network of figure 4, it is easy and very fast to determine performance measures of components through Mean Value Analysis (MVA) algorithm described in [8]. The arrival theorem in mixed product-form queuing network states that when request arrives at a function in open network it will see the number of requests in equilibrium, while request arrives in closed network will see the number of requests in equilibrium with one request less in the network [9].

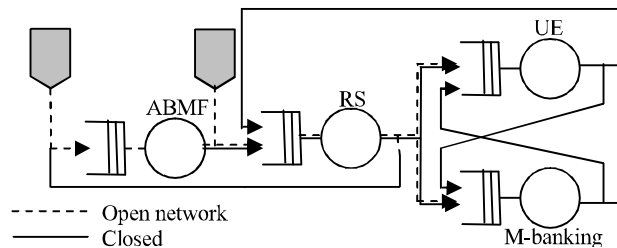


Figure 4. Mixed product-form network model of M-SaaS recharging scheme (non-absorbed)

However, MNO's infrastructures are shared by many customers, the fact which makes ABMF, RS, and M-banking to arbitrarily handle large number of requests. Due to their involvement in other MNO's activities, the determination of performance measures of open queue network becomes much complex and consume much memory due to prospective large number of requests. The best approach to relieve the memory is to transform the open part to a suitable constructed closed network [10]. As shown in figure 5, this is done by replacing external world of open network by a $-G/l$ loop as described in [7].

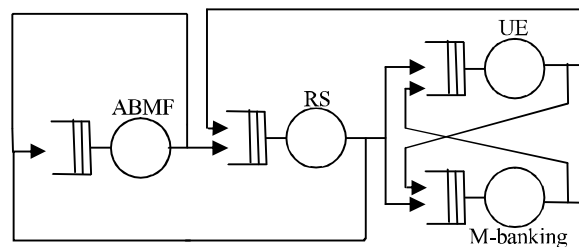


Figure 5. A closed product-form of figure 4

With closed product-form queuing network, the performance metrics can easily approximated by considering solution given in [11], which suggested an approximation of MVA for single server queuing networks based on the following idea: Starting with an initial network load, $\bar{L}_i(L)$, for mean number of requests at function i for a given population-vector L , make an estimate of the mean number

of requests for a population vector $(L-1)$ with an assumption that requests are equally distributed over the whole network.

Obvious, the problem with this MVA approach is how to estimate the mean value $\bar{L}_i(L-1)$ by $\bar{L}_i(L)$. The problem lies on definition of closed network whereby instead of mean number of requests in node i to reduced by one, it remain nearly the same. In this case a reasonable explanation is given in [7], which suggests based on large network population that it is possible to treat $\bar{L}_i(L-1) = \bar{L}_i(L)$ in initial stage only.

Thus, given the functional service rates and visit ratio of RR, the fact that functions involved in M-SaaS recharging scheme also serves large number of requests makes the MVA algorithm to be described in following sequences of steps:

Step 1: Initialization: Based on assumption that requests are initially equal distributed over the whole network of functions, for $i = 1, \dots, N$

$$\bar{L}_i(L) = \frac{L}{N} \quad (1)$$

Step 2: For all functions, $i = 1, \dots, N$ compute the estimated values for the mean number of requests for the population vector $(L - 1)$ such that:

$$\bar{L}_i(L-1) = \frac{(L-1)}{L} \bar{L}_i(L) \quad (2)$$

In initial iteration the above equation can be replaced by equation (1) such that:

$$\bar{L}_i(L-1) = \frac{(L-1)}{N} \quad (3)$$

Step 3: The mean response time, throughput and distribution of requests at equilibrium are iteratively computed until the check condition stated in step (4) satisfied. The iteration given such that:

(a) Estimate the response time using formulae for closed network derived in [7] such that

$$\Psi_i = \frac{1}{\mu_i} \cdot (1 + \bar{L}_i(L-1)) \quad (4)$$

(b) Determine the general throughput (ρ) based on closed network computation formulae such that

$$\rho = \frac{L}{\sum_{i=1}^N v_i \Psi_i} \quad (5)$$

(c) Determine distributed mean number of requests in each function at equilibrium such that:

$$\bar{L}_i(L) = \rho \Psi_i v_i \quad (6)$$

Step 4: Check the stopping condition: This step intend to check whether there significant changes in $L_i(L)$ between successive iteration, otherwise the algorithm should execute next iteration by go back to step (2). To check for significance change of $L_i(L)$ use the following formulae

$$\max_i |\bar{L}_i^{(n)}(L) - \bar{L}_i^{(n-1)}(L)| < \varepsilon \quad (7)$$

where n and $n-1$ denotes two successive iterations and ϵ denotes favorable check point bound set to be 0.25.

3.2. Estimation of Model Parameters

Consider the MVA algorithm stated in previous subsection, the algorithm requires three raw input parameters: initial mean population of RR in the network (L), visit-ratio (v_i) and service times (μ_i). Estimation of L is more crucial because iterating through varied L help to deduce behavior of recharging scheme over varied network load. The v_i and μ_i can be obtained by monitoring traffic of RR in possible operating environment.

3.2.1. Visit-ratio

The visit-ratio (v_i) is the average number of times that i^{th} function invoked through RR lifetime. The collective behavior of v_i reflects service demand of respective request within the system. Therefore, let λ_i denotes the arrival rate of RR to i^{th} function, the average arrival rate in a network of k functions is given such that:

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^K \lambda_j \mu_{j,i}$$

which calculated by adding arrival rate of RR trigger at SBCF ($\lambda_{0,i}$) and arrival rates from neighboring peers. $\mu_{j,i}$ is the rate that i^{th} function serves arrived RR from j^{th} function.

Based on overall arrival rate (λ) of recharging requests, the visit ratio (v_i) of recharging requests to i^{th} function, referred to as relative arrival, can be obtained such that

$$v_i = \frac{\lambda_i}{\lambda} \quad (8)$$

3.2.2. Service time

The core challenge in this development lies on estimation of service rate of intermediate functions. Let $\pi(f_1, \dots, f_N)$ denotes the steady-state transient vector of M-SaaS recharging scheme, where f_i represents functions involved in respective state. The system of equation used to estimates transient matrix π is derived and proved in [7] such that:

$$0 = \pi Q \quad (9)$$

where Q is infinitesimal generator matrix, which in M-SaaS recharging scheme is depends on service time (μ) because state transition within the scheme depends on service of functions.

The transient matrix π is estimated based on four non-absorbed states, described in section 3.1 and figure 5, results into matrix of 1 by 4. Each time RR is served in i^{th} state, the rest j^{th} states ($j \neq i$) have to wait or participate in other activities until a transition occur. Therefore, the transient vectors of such behavioral system are derived such that: $\{(1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1)\}$. Figure 6 shows the transition diagram that includes transient vector and corresponding Q .

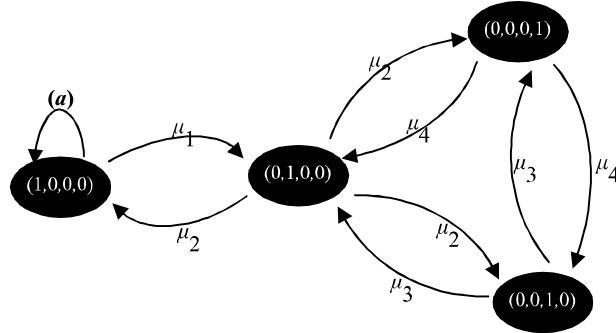


Figure 6. Recharging scheme transition diagram

Based on transition flux-in and flux-out, the corresponding global balance equations are derived with reference to equation (9) such that:

$$0 = \pi(1,0,0,0) \mu_1 - \pi(0,1,0,0) \mu_2 \quad (10)$$

$$0 = \pi(0,1,0,0) \mu_2 + \pi(0,1,0,0) \mu_2 + \pi(0,1,0,0) \mu_2 - \pi(1,0,0,0) \mu_1 - \pi(0,0,1,0) \mu_3 - \pi(0,0,0,1) \mu_4 \quad (11)$$

$$0 = \pi(0,0,1,0) \mu_3 + \pi(0,0,1,0) \mu_3 - \pi(0,1,0,0) \mu_2 - \pi(0,0,0,1) \mu_4 \quad (12)$$

$$0 = \pi(0,0,0,1) \mu_4 + \pi(0,0,0,1) \mu_4 - \pi(0,1,0,0) \mu_2 + \pi(0,0,1,0) \mu_3 \quad (13)$$

Note that the flux-in and flux-out due to transition (a) in figure 6 cancel-over each other, which we found useful not to include in equation (10). By rearrange and summarize equations (10), (11), (12) and (13), we can represent them by single expression such that:

$$\mu_i = \frac{1}{m_i \pi_i} \sum_{j=\forall m} \pi_j \mu_j \quad (14)$$

where m_i denotes the number of neighbors of i^{th} function.

Considering the normalization condition of transient matrix (that gives $\sum \pi = 1$), and representation of service time in terms of service demand elaborated in [7], then equation (14) is reduced such that:

$$\mu_i = \frac{1}{m_i \pi_i} \sum_{j=\forall m} \lambda_j v_j \quad (15)$$

The possible recurring visit of RR into i^{th} function provides possibility to determine transient matrix based on its relation to visit ratio such that $\pi_i = 1/\lambda_i v_i$. Replacing this representation into equation (15) we get:

$$\mu_i = \frac{\lambda_i v_i}{m_i} \sum_{j=\forall m} \lambda_j v_j \quad (16)$$

4. Numerical Example and Analytical Explanation

4.1. Metrics Derivation

To investigate performance measures of M-SaaS recharging scheme our study establishes numerical examples, whose figures are to be derived based on underlined algorithm and relevant studies. The sequence flow of M-SaaS recharging scheme elaborated in section 2.1 and its analytical model in section 3.1 gives insight on how involved components work on RR. By observing RR through analytical model based on steps of the sequence flow, one can deduce number of recurring visits of RR in each function. Table 1 shows possible visits made by RR throughout its lifespan. Based on number of visits, we derive corresponding visit ratio (v_i) by considering equation (8).

Table 1. Visit ration of RR in M-SaaS recharging scheme

	ABMF ($i = 1$)	RS ($i = 2$)	M-banking ($i = 3$)	UE ($i = 4$)
Number of visits (λ_i):	2	2	4	3
Visit ratio (v_i)	0.1818	0.1818	0.3636	0.2727

Service rate relatively implies the rate at which RR served by components of M-SaaS recharging scheme. According to subsection 3.2.2, the rate is associated with number of neighbors and recurring visits of RR to the components. Based on this association, number of visits, visit-ratio, and relevant values of m_i obtained by referring to figure 6, we develop estimation of service rate based on equation (16) as shown in table 2.

Table 2. Service rates of components in M-SaaS recharging scheme

	ABMF ($i = 1$)	RS ($i = 2$)	M-banking ($i = 3$)	UE ($i = 4$)
Number of Neighbors (m_i)	1	3	2	2
* Service rate ($1/\mu_i$)	7.5625	3.1299	1.1637	1.3447

A study by Banerjee S., et. al. in [12] shed light on number of requests in SaaS session. The study observed that majority of SaaS sessions exhibit between 76 and 80 requests. Although this dissemination could not reflect the load in M-SaaS recharging scheme but it gives insight about possible value of L . In this analysis we keep number of RR (L) lower than total requests in SaaS session disseminated in [12] due to a reason that obviously only one RR and/or retransmission of one RR will be generated for single user. To analyze performance requests due to varying load we choose $L = 1$, and alternate through a load of 2, 3, 4 and 5.

4.2. Numerical Computation

Given the visit ratio and service rates deduced in previous subsection, the numerical computation based on these metrics is developed through established MVA algorithm. Since the computation with $L=1$ is simple, our example considers $L=2$, leaving the rest to interested readers, who may verify answers with figures in table 3.

$L=2$:

Step 1: Initialization

$$\bar{L}_1(L) = \bar{L}_2(L) = \bar{L}_3(L) = \bar{L}_4(L) = \frac{L}{N} = 0.5$$

1st Iteration:

Step 2: Estimation of initial $\bar{L}_i(L-1)$ obtained by considering equation (3) such that

$$\bar{L}_1(L-1) = \frac{L-1}{L} \cdot \bar{L}_1(L) \Rightarrow \bar{L}_1(L-1) = \frac{L-1}{N} = 0.25$$

Likewise

$$\bar{L}_2(L-1) = \bar{L}_3(L-1) = \bar{L}_4(L-1) = 0.25$$

Step 3: MVA algorithm to obtain relevant response time, throughput and number of requests in each tiers at equilibrium performed in following sub-steps:

(a) Response time obtained by consider equation (4) such that

$$\Psi_1 = \frac{1}{\mu_1} \cdot (1 + \bar{L}_1(L-1)) = 9.4531,$$

Likewise

$$\Psi_2 = \frac{1}{\mu_2} \cdot (1 + \bar{L}_2(L-1)) = 3.9124, \quad \Psi_3 = \frac{1}{\mu_3} \cdot (1 + \bar{L}_3(L-1)) = 1.4546,$$

$$\Psi_4 = \frac{1}{\mu_4} \cdot (1 + \bar{L}_4(L-1)) = 1.6809$$

(b) Throughput obtained by considering L , visit-ratios, throughput and equation (5) such that

$$\rho = \frac{L}{\sum_{i=1}^N v_i \Psi_i} = \frac{2}{1.7186 + 0.7112 + 0.5289 + 0.4584} = 0.5853$$

(c) Mean number of request at equilibrium obtained by replacing above parameters into equation (6) such that:

$$\bar{L}_1(L) = \rho \Psi_1 v_1 = 1.0059, \quad \bar{L}_2(L) = \rho \Psi_2 v_2 = 0.4163, \quad \bar{L}_3(L) = \rho \Psi_3 v_3 = 0.3096,$$

$$\bar{L}_4(L) = \rho \Psi_4 v_4 = 0.2683$$

Step 4: Check for stop condition such that

$$\max_i |\bar{L}_i^{(n)}(L) - \bar{L}_i^{(n-1)}(L)| \Rightarrow \max_i |\bar{L}_i^{(1)}(L) - \bar{L}_i^{(0)}(L)| \Rightarrow |1.0059 - 0.5| = 0.5059 < 0.25$$

Since stop condition is not satisfied we move to the next iteration:

2nd iteration:

Step 2: Estimation of $\bar{L}_i(L-1)$ obtained by considering equation (2) such that

$$\bar{L}_1(L-1) = \frac{L-1}{L} \cdot \bar{L}_1(L) \Rightarrow \frac{2-1}{2} * 1.0059 = 0.5029,$$

Likewise

$$\bar{L}_2(L-1) = 0.2081, \quad \bar{L}_3(L-1) = 0.1548, \quad \bar{L}_4(L-1) = 0.1341$$

Step 3: MVA algorithm to obtain relevant response time, throughput and number of requests in each tiers at equilibrium performed in following sub-steps:

(a) Response time obtained by consider equation (4) such that

$$\Psi_1 = 11.3659, \Psi_2 = 3.7814, \Psi_3 = 1.3438, \text{ and } \Psi_4 = 1.5251$$

(b) Throughput obtained by considering L , visit-ratios, throughput and equation (5) such that

$$\rho = \frac{L}{\sum_{i=1}^N v_i \Psi_i} = \frac{2}{2.0663 + 0.6875 + 0.4886 + 0.4159} = 0.5467$$

(c) Mean number of request at equilibrium obtained by replacing above parameters into equation (6) such that:

$$\bar{L}_1(L) = 1.1297, \bar{L}_2(L) = 0.3758, \bar{L}_3(L) = 0.2671, \text{ and } \bar{L}_4(L) = 0.2274$$

Step 4: Check for stop condition such that

$$\max_i |\bar{L}_i^{(n)}(L) - \bar{L}_i^{(n-1)}(L)| \Rightarrow \max_i |\bar{L}_i^{(2)}(L) - \bar{L}_i^{(1)}(L)| \Rightarrow |1.1297 - 1.0059| = 0.1238 < 0.25$$

Since the stop condition is satisfied then the iteration stops and we can now consider obtaining other performance measures as follows:

- Average response time is deduced from response time in last iteration (2nd iteration), such that:

$$\bar{\Psi} = \frac{\sum_{i=1}^N \Psi_i^n}{N} = \frac{11.3659 + 3.7814 + 1.3438 + 1.5251}{4} = 4.5040$$

- The throughput of each tier is computed based on overall throughput such that:

$$\rho_1 = \rho \cdot v_1 = 0.0994, \rho_2 = \rho \cdot v_2 = 0.0994, \rho_3 = \rho \cdot v_3 = 0.1988 \text{ and } \rho_4 = \rho \cdot v_4 = 0.1491$$

- Then, the utilization of each tier is computed such that:

$$\delta_1 = \frac{\rho_1}{\mu_1} = \frac{0.0994}{0.1322} = 0.7519, \delta_2 = 0.3111, \delta_3 = 0.2313 \text{ and } \delta_4 = 0.2005$$

Through the same procedures we compute performance measures when initial total requests are 1, 3, 4 and 5. The results are summarized in table 3.

Table 3. Values of performance measures through iteration on MVA

		Initial total requests				
functions		1	2	3	4	5
No. of requests at equilibrium (\bar{L}, L)	ABMF	0.5029	1.1297	1.9073	2.7069	3.6649
	RS	0.2081	0.3758	0.4842	0.5851	0.6101
	M-banking	0.1548	0.2671	0.3308	0.3870	0.3969
	UE	0.1341	0.2274	0.2777	0.3210	0.3281
Response Time (Ψ_i)	ABMF	7.5625	11.3659	16.6641	21.7714	28.9505
	RS	3.1299	3.7814	4.2307	4.7058	4.8191
	M-banking	1.1637	1.3438	1.4451	1.5564	1.5677
	UE	1.3447	1.5251	1.6176	1.7214	1.7278
Throughput (ρ_i)	ABMF	0.0665	0.0994	0.1145	0.1243	0.1266
	RS	0.0665	0.0994	0.1145	0.1243	0.1266
	M-banking	0.1330	0.1988	0.2289	0.2487	0.2532
	UE	0.0998	0.1491	0.1717	0.1865	0.1899
Utilization (δ_i)	ABMF	0.5029	0.7516	0.8656	0.9403	0.9574
	RS	0.2081	0.3111	0.3582	0.3891	0.3962
	M-banking	0.1548	0.2313	0.2664	0.2894	0.2946
	UE	0.1341	0.2005	0.2309	0.2508	0.2553

4.3. Numerical Results Analysis

4.3.1. Utilization over growing requests

Figure 7 plots a graph of utilization (ρ_i) of components involved in M-SaaS recharging scheme over increase recharging requests seen by RR when arrive into the scheme. The graph depicts polynomial increase of utilization as the load grows. Furthermore, each curve observed to show converging behavior to own upper limit. This evidenced that the increase in components utilization due to increasing in recharging requests is characterized with upper/maximum limit. The design implication of this characteristic is that it would be possible to control the utilization of components by allocating relevant static memory and processing capacity that reflect upper bound utilization.

On the other hand, the relative lower utilization of user equipment (UE) depicted in figure 7, evidenced low involvement of UE in M-SaaS recharging scheme. This indicates relative minimum human intervention, which implies a degree of automation of M-SaaS recharging scheme. The fact that there would be minimal human intervention reduces possibility of human error and increase accuracy of the proposed model.

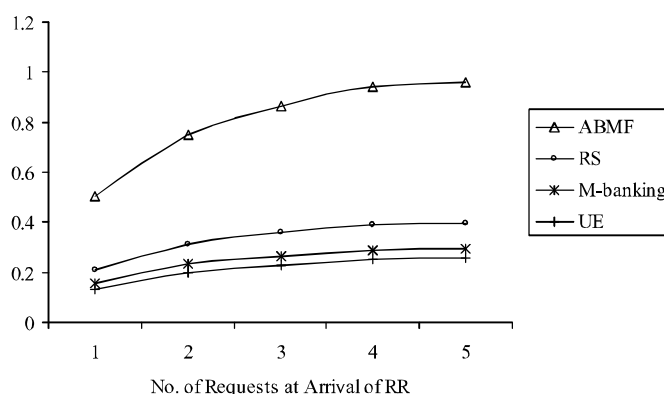


Figure 7. Utilization of Components involved in M-SaaS Recharging Scheme

4.3.2. Throughput over increase initial load

Figure 8 plots a graph of throughput (δ_i) of components in M-SaaS recharging scheme over increase number of recharging requests seen by tagged RR when arrive into the scheme. The graph depicts an increase of throughput as load due to recharging requests grow. This evidenced that the throughput of involved components effectively respond to increasing load, the behavior that lead to speed up of processing of requests and so as to reduce memory burden.

However, the relative higher throughput shown by M-banking and user equipment (UE) evidenced that in order for the scheme to be efficiency, the two components need to respond promptly whenever receive recharging request. This requirement implies that designers of M-SaaS need to consider a possibility of inducing priority-based queuing discipline into operation of these components so as to fasten their response.

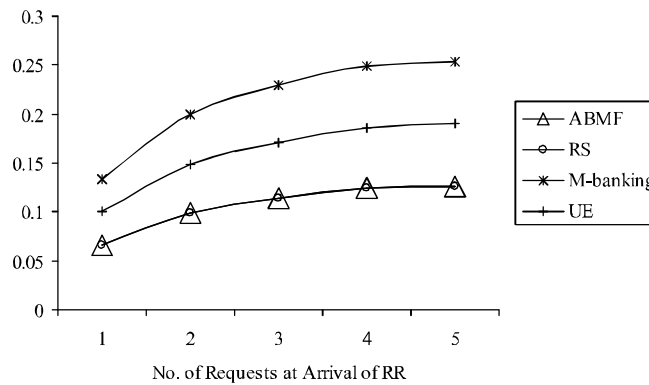


Figure 8. Throughput of components involved in recharging scheme

4.3.3. Effect of recharging requests distribution at equilibrium

Figure 9 plots a graph of distribution of recharging-requests at equilibrium over increase number of recharging requests seen by tagged RR when arrive into M-SaaS recharging scheme. The graph depicts an exponential increase of distribution to ABMF as the load grows, indicating unmanageable characteristics. This characteristic is not healthy to the scheme due to the fact that ABMF in collaboration with SBCF plays major role in online charging of session-based activities in MNO. If ABMF becomes unmanageably overloaded then its participation into other MNO charging activities won't be efficiency. Therefore, our team is currently engage in further study to device useful means of adjusting this behavior.

On the other hand, the corresponding curves of RS, M-banking and UE show converging behavior to own upper bounds. This shows a stable and manageable distribution behavior at equilibrium, which implies that designers can even set static processing and memory capacity for the system to operate effective.

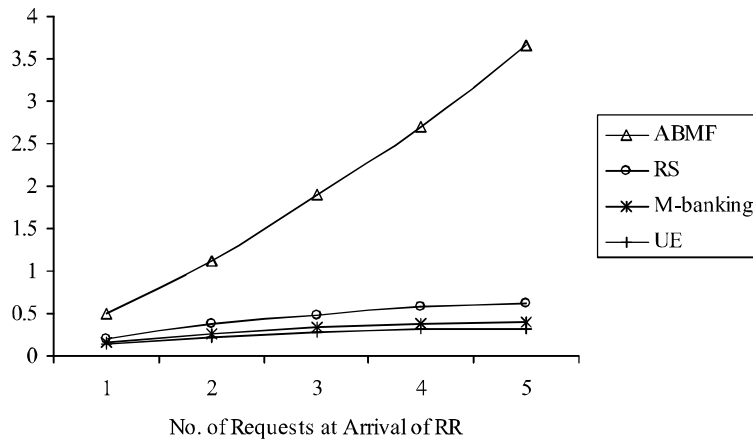


Figure 9. Mean requests distribution at equilibrium

4.3.4. Mean response time over growing load

Figure 10 plots a graph of average mean-response time (Ψ) of M-SaaS recharging scheme over increase number of recharging requests seen by RR when arrive into the scheme. The graph depicts a slight exponential increase as the load increases. This implies that the recharging scheme responds relatively fast under low load but the response speed deteriorates over growing load. This might have negative effect to the scheme amid the sharing nature of the components to MNO activities. Thus, it is advisable to regulate generation of recharging requests into the scheme to certain threshold maximum point.



Referring to elaboration of usefulness criteria for M-SaaS recharging scheme in section 2.2. It has been argued that mean response time should be less than session window (ΔS). Based on the result of figure 10, the design of M-SaaS recharging scheme should maintain shorter ΔS due to the fact that the shorter ΔS the less recharging requests would be generated because of low tariff (relatively). This is suffice to comment that keeping shorter ΔS may help to regulate generation of recharging requests into the scheme.

5. Conclusion

In this study we analyzed performance measures of M-SaaS recharging scheme, which together with M-SaaS credit-reservation scheme, makes M-SaaS model. The Analysis establishes analytical model based on sequence flow of M-SaaS recharging scheme that lead to development of its network queuing model and estimation of useful parameters.

Although the numerical result shows healthy utilization and sense of response on throughput of involved components but exponential increase of distribution of load by ABMF curve at equilibrium suggest unhealthy characteristics because of sharing nature of this components in MNO activities. Further study to find useful means to reduce such burden is underway. Also, the exponential increase in average response time suggests a need to regulate generation of recharging requests into the scheme. The immediate measure to keep fewer recharging requests is to maintain shorter session window (ΔS), which in-turn shorten the Δt_R . It has been disseminated hereinto that shorter Δt_R could cause recharging window to hardly match with total response time and lead to force termination of session. Therefore, using the measure of shortening ΔS requires cautious design to accommodate reasonable recharging window (Δt_R).

Currently, our team is involved in studying a useful approach to solve exponential increase on load distribution on ABMF at equilibrium. In our humble opinion, the approach should be intelligent enough not only to reduce the burden on ABMF but also to distribute detached load equally to other components without overloading single component and cause a circle of inefficiency.

6. Reference

- [1] [Anirban Kundu, Chandan Banerjee, Priya Saha, "Introducing New Services in Cloud Computing Environment", JDCTA, Vol. 4, No. 5, 2010, pp. 143 - 152.](#)
- [2] [Abhijit Dubey and Dilip Wagle, "Delivering Software as a Service", Web Exclusive Report, McKinsey & Company, May 2007, accessed from \[www.mckinsey.de/\]\(http://www.mckinsey.de/\) on December 2010.](#)
- [3] [Lianru Liu, Meina Song, Xiaoxiang Luo, Haoping Bai, Shangbin Wang, and Junde Song, "An Implementation of the Online-payment Platform based on SaaS", Proceeding of IEEE 2nd Symposium on Web Society \(SWS\), October 2010, pp. 658 - 662.](#)
- [4] [Jim Rosenberg, "Observation: Few Poor and Unbanked People have Begun using Branchless Banking for Financial Services", accessed from <http://technology.cgap.org/2008/07/03/> on October 2010.](#)
- [5] [Joseph Cosmas Mushi, Guan-zheng Tan, Cheruiyot Wilson, "Enhanced SaaS Delivery in Developing economy", International Journal of Convergence Information Technology, Vol. 6, No. 4, April 2011, pp. 70 - 80.](#)
- [6] [Joseph Cosmas Mushi, Guan-zheng Tan, Felix Musau, Cheruiyot Wilson, "Modeling M-SaaS Delivery Model with Threshold-based Credit Recharging using M-banking", Proceeding of the 3rd IEEE International Conference on Computer Research and Development, Vol.2, March 2011, pp. 307-311.](#)
- [7] [Gunter Bolch, Stefan Greiner, Hermann de Meer, Kishor S. Trivedi, "Queuing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications – 2nd edition", John Wiley & Sons, inc., USA, 2006.](#)
- [8] [Lam, S.S, "A Simple Derivation of the MVA and LBANC Algorithms from the Convolution Algorithm", IEEE Transaction on Computer, Vol. C-32, Issue 11, November 1983, pp. 1062 – 1064.](#)
- [9] [M. Reiser and S. S. Lavenberg, "Mean-Value Analysis of Closed Multichain Queuing Networks", Journal of the ACM \(JACM\), Vol.27, Issue 2, April 1980, pp. 313 - 322.](#)
- [10] [Forest Baskett , K. Mani Chandy , Richard R. Muntz , Fernando G. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers", Journal of the ACM \(JACM\), Vol. 22, Issue 2, April 1975, pp.248 - 260.](#)
- [11] [Yonathan Bard, "Some Extensions to Multiclass Queueing Network Analysis", Proceedings of the 3rd International Symposium on Modeling and Performance Evaluation of Computer Systems, North-Holland Publishing Co. Amsterdam, Netherlands, 1979, pp. 51 – 62.](#)
- [12] [Sean Banerjee, Hema Srikanth, Bojan Cukic, "Log-based Reliability Analysis of Software as a Service \(SaaS\)", IEEE 21st International Symposium on Software Reliability Engineering \(ISSRE\) , San Jose, CA, November 2010, pp. 239 – 248.](#)