

유무선 통합 망에서 대량호 지능망 서비스의 성능 및 자원 최적화를 위한 동적 큐 관리자

정희원 최한옥*, 안순신*

A Dynamic Queue Manager for Optimizing the Resource and Performance of Mass-call based IN Services in Joint Wired and Wireless Networks

Han-Ok Choi*, Sunshin An* *Regular Members*

요 약

본 논문은 유무선 통합망 환경하에서 대량호에 근거한 지능망 서비스의 호 완료율을 증가시키고 자원을 최적화하기 위한 정보 흐름 및 총괄서비스 로직을 설계한다. 이 로직을 구현하기 위하여 지능망 서비스제어기에 호대기 서비스 특징을 적용하기 위한 동적 큐 관리자를 제시하는데, 이것을 유선 가입자뿐만 아니라 이동 단말을 소유한 무선 가입자에게도 적용하기 위해 홈위치등록기와 서비스제어기간에 해당 서비스 가입자들의 이동성을 큐 관리자에게 통보하기 위한 서비스 등록 여부 프래그를 관리한다. 따라서, 이동 서비스 가입자가 로밍함에 따른 동적 서비스 그룹핑을 수행하고 M/M/c/K 큐잉 모델에 근거한 큐 길이를 동적으로 관리하여 자원을 최적할 수 있는 동적 큐 관리 메커니즘을 제안한다. 동적 큐 관리자에 의해 할당된 큐 길이를 결정하기 위해 서비스 증가율에 따른 가입자 단말 수와 실패율간의 관계 및 그에 따른 큐에서의 적정 대기 시간을 시뮬레이션 및 분석 한다. 또한, 동적 큐 관리자를 구성하는 내부 서비스 로직과 SIB들간의 상호동작 및 자료 구조를 설계 및 구현한다.

ABSTRACT

This paper proposes enhanced designs of global service logic and information flow for the mass-call based IN service, which increase call completion rates and optimize the resource in joint wired and wireless networks. In order to have this logic implemented, we design a Dynamic Queue Manager (DQM) applied to the call queuing service feature in the Service Control Point (SCP). In order to apply this logic to wireless service subscribers as well as wired service subscribers, the service registration flags between the Home Location Register (HLR) and the SCP are managed to notify the DQM of the corresponding service subscribers' mobility. Hence, we present a dynamic queue management mechanism, which dynamically manages the service group and the queue size based on M/M/c/K queueing model as the wireless subscribers roam the service groups due to their mobility characteristics. In order to determine the queue size allocated by the DQM, we simulator and analyze the relationship between the number of the subscriber's terminals and the drop rate by considering the service increment rate. The appropriate waiting time in the queue as required is simulated according to the above relationship. Moreover, we design and implement the DQM that includes internal service logic interacting with SIBs (Service Independent building Blocks) and its data structure.

* 고려대학교 전자공학과 네트워크 연구실(hochol@dsys.korea.ac.kr)
논문번호 : 99482-1203, 1999년 12월 3일

1. Introduction

The design of Intelligent Network (IN) follows a simple principle: separation of service-specific software from basic call processing. The IN reduced a great deal of this dependency by moving service-specific software into specialized nodes called Service Control Points (SCPs). Basic call processing is performed in the switches, named Service Switching Points (SSPs). The communication between SSPs and SCPs is done through a channel called Common Channel Signaling number 7 (CCS7). The advent of IN reduced the time frame of the introduction of a new service to a few months. That is, the IN architecture allows a variety of different services to be provided to customers independent of the underlying network technologies. The IN is turning the network into a programmable entity and thus providing an open platform, which supports the uniform creation, control, and management of services beyond the basic telephone services in the telecommunication environment [1]. IN Capability Set 2 (CS-2) is the second standardized stage of the IN as an architectural concept for telecommunication services, service management services and service creation services [3][4][5][6]. The success of IN concepts steered their application in the mobile communications environment; examples of such applications are Wireless Intelligent Network (WIN) and Customized Applications for Mobile network Enhanced Logic (CAMEL) [11].

The WIN is a network which supports the use of IN capabilities to provide advanced network services in the mobile environment while minimizing backwards compatibility issues that could arise from modifying the existing TIA/EIA-41 operations. Certain service features may also be implemented using the service control or service data functions associated with service features through the distributed network among the Mobile Switching Center (MSC),

Home Location Register (HLR), Visitor Location Register (VLR), Service Control Point (SCP), Intelligent Peripheral (IP), or other network elements [8].

Thus such a SCP needs to support various protocols and allows a smart combination of IN and mobile services to provide subscribers with reasonable comfort and familiarity as they move between systems.

In other words, the IN concept could be applied to any kind of network; Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN), and wireless networks. It thus takes more and more importance in wired and wireless networks providing respectively mobility and flexibility [12][13]. Moreover, research on how IN could be efficiently applied to wireless networks to enhance services for these networks has been made rapid progress.

As networks move toward advanced INs, more services could be provided, causing increase in the number of users. Moreover, due to the development of wireless networks, the number of mobile subscribers is increasing, too. New services and enhanced mobility features motivated a service paradigm based on joint wired and wireless networks, service offerings can create interesting opportunity on the commercial aspect. As a matter fact, this concept always implies service and/or commercial objectives, regardless of the technical and operational solutions adopted for its realization [14]. In particular, traffic engineering for wired networks has been successfully developed based on a communication paradigm which has included regulated operation environment and predictable service quality. These assumptions are being changed more and more with the increasing popularity of wireless communications services and deregulated operation. In addition, more specifically mass call processing in a specific time for wireless service subscribers as well as wired service subscribers might be taken into account.

However, the IN service providers have been

trying to estimate how many queues and subscribers' terminals could be needed to guarantee call completion for their service subscribers by only their practical experience. Even if so, they have applied the static queue mechanism focused on a wired subscriber.

In this paper, we propose the dynamically queuing service feature for IN service enhancement according to the increment of IN service subscribers and their mobility. Hence we implement a DQM for assigning queue sizes to increase call completion rates for service enhancements in wired and wireless intelligent networks.

As shown in Figure 1, the Personal Communication eXchange (PCX) acts as a switching point for both wired networks and wireless networks. The HLR has service registration flags for each mobile subscriber to notify the SCP of the subscriber's terminal mobility and other information about the service profile. Moreover, the SCP has a DQM to manage queues according to the regional groups of subscribers.

The main features of the mechanism are as follows: Firstly, it can be used for IN service enhancement in wireless networks as well as in wired networks. Secondly, it has a dynamic queue management to guarantee mass call processing to handle large number of concentrated requests on the service with a single number during a certain time. Thirdly, it has a call distribution algorithm available to route calls in a fair manner. Finally, it adjusts dynamically queue sizes and service subscribers groups according their mobility.

This paper is organized as follows. In the first two sections, we describe an applied system model, call information flow to guarantee the mass call processing on wired and wireless networks, and location update information flow relating to subscriber's mobility. In the remaining sections of the paper, we present dynamic queue manager which include internal service logic, data structure, and dynamic management of queue size

based on through the M/M/c/K queuing theory considering the mobility of servers. We simulate our proposed model to get an appropriate queue size and waiting time in the dynamic queue according to the service increment rate and subscriber's terminals mobility with each drop rate by agreement on between service subscribers and service providers in advance. Finally, we conclude our paper with a summary and outlook.

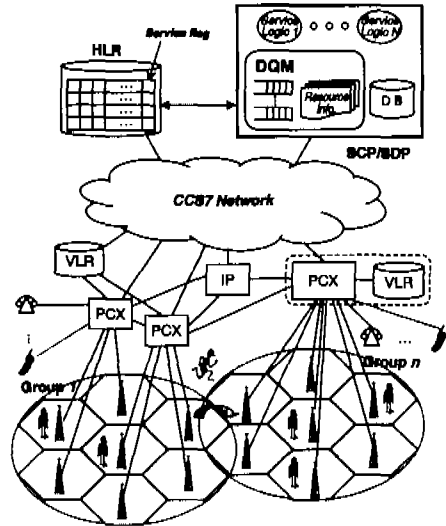


Fig. 1 An applied system model

II. Call Information Flow

In order to accommodate a large number of simultaneous calls to a specific service number - which allows a subscriber with two or more terminating lines (i.e., called parties) in any number of locations to have a single directory number - during a certain time, the DQM enables a service user (i.e., calling party) encountering busy signals to have the call completed when they become available, without having to make a new call attempt [1].

Global Service Logic (GSL) which support the overall service processing for the queue service feature applied service enhancement views the basic call process as a single resource, that offers

a set of operations. Based upon such a view of IN services, the following are identified as necessary interactions between GSL and Basic Call Process (BCP) [5]. As shown in Figure 2, the GSL is decomposed into appropriate Service Independent Building Block (SIB) chains. The BCP is responsible for providing basic call connectivity between parties in the network. The BCP can be viewed as a specialized SIB operations.

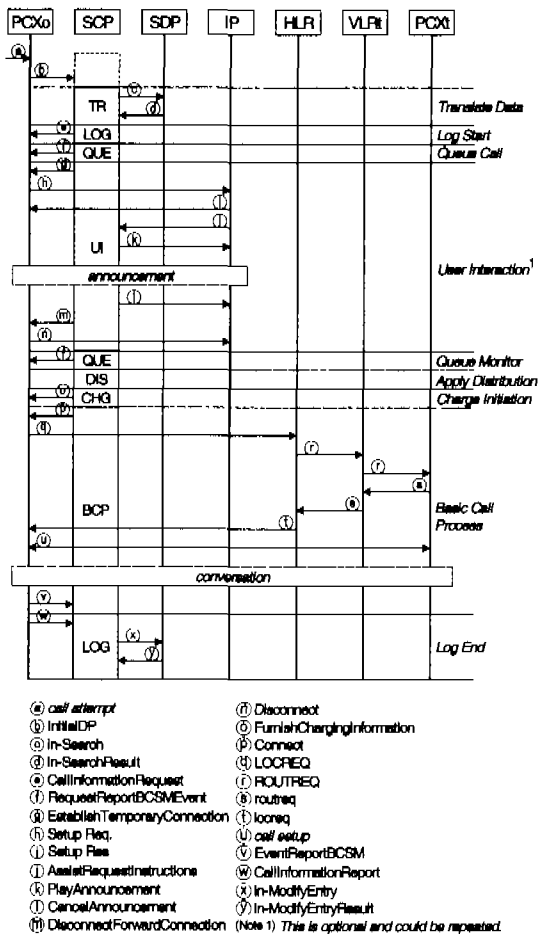


Fig. 2 Call Information flow diagram

The information flow for providing mass call connectivity with appropriate disposition between parties in the networks, where a service user is connected to a wireless IN service subscriber's terminal, is shown in Figure 2. It constitutes the

actions of the resource manager, dealing with resource availability and queue handling, and the sending and receiving of messages, which are described by the distributed service logic [6][7].

When the calling party (i.e., service user) sends an indication that it wishes to originate a call (a), the originating PCX (PCXo) analyzes the dialed service number. On satisfying the origination trigger criteria, the PCXo sends the InitialDP operation (b) to get the service subscriber's destination number based on time dependent routing from the corresponding SCP through the Service Data Point (SDP). The Translate Data SIB (c,d) is invoked at that time. The Log Start SIB directs the PCXo to record the detailed information for each call into a file using the CallInformationRequest operation (e). The collected information may be used by the network management services (e.g., statistics, etc.) and not by call-related services. During the queuing of the call, the queue will be checked for available queue space. The status of resources can be determined by monitoring the BCSM events. The Queue SIB sends the RequestReportBCSMEvent (f), to arm the "user abandon" Detection Point (DP) as an Event Detection Point (EDP) to ensure that calling party is removed from the queue on abandon.

If all the service subscriber's terminals in the group are busy, (g)-(n) are processed so that the service logic stores the relevant call in the queue with playing announcements to the queued call through the User Interaction SIB. If one or more available subscriber's terminal exist in the group, (g)-(n) and the secondary (f) are not processed but the Apply Distribution SIB is immediately performed. The Distribution SIB could distribute calls to its different terminals based on the lowest service distribution percentage algorithm. If there is no resource available to be allocated to the queue, or if the queue waiting time is expired, the fact that the call cannot be processed any more is notified to the service user. In order to announce the service user, firstly, the SCP passes

the EstablishTemporaryConnection operation to the PCXo (g) so that the PCXo connects a bearer channel to the Intelligent Peripheral (IP) through (h) and (i) operations. Then the IP sends the AssistRequestInstruction operation (j) to the SCP, and the SCP instructs the IP to play the appropriate announcement through the PlayAnnouncement operation (k). If an event that inevitably stops the announcement occurs (i.e., one of subscriber's terminals becomes available, or the timer expired), the SCP instructs the IP to cancel the announcement (l), the PCXo to disconnect with the IP (m).

The FurnishChargingInformation (o) operation in the Charge Initiation SIB gives some charging information to the PCXo, to enable it to generate an appropriate billing record for the current call. The generated record at the end of the call may be sent by the PCXo to some billing center. This operation may be invoked several times during a call.

The SCP responds with the Connect operation (p) including a destination number, then the PCXo sends the LOCREQ (q) operation to the HLR. The HLR determines the PCXt for the called terminal and sends the ROUTREQ operation (r) to the PCXt through the VLRt. The PCXt assigns a (Temporary Local Directory Number (TLDN) and returns the TLDN as the destination digits in the routreq operation (s) to the HLR through the VLRt. The HLR returns the destination digits to the PCXo in the locreq operation (t). The PCXo establishes a voice path to the PCXt using existing interconnection protocols and the routing information specified in the locreq operation.

When the call is successfully connected to the serving PCX (PCXt) because of the call setup (u), the conversation between the service user and the chosen subscriber's terminal is occurred. When the conversation is over, the PCXo reports the fact to the SCP (v). Then the SCP performs the Log End SIB (w-y) so that the call information is stored in the SDP.

After the call setup between parties, further calls to service logic can be performed when armed detection points are encountered and appropriate criteria are satisfied [9][10].

III. Location Update Information Flow

Figure 3 depicts the information flow for reorganizing subscribers groups according to their mobility. When the new serving PCX (PCXn) determines that a subscriber's terminal has been registered in its service group area, it forwards the REGNOT operation (a) to the HLR associated with the subscriber's terminal through the new serving Visitor Location Register (VLRn).

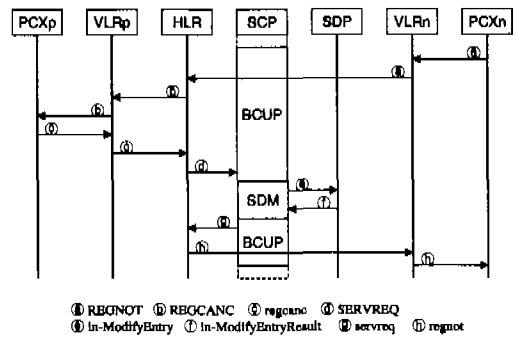


Fig. 3 Location update information flow diagram

After determining that the roaming subscriber's terminal has left the service group area in which it had been registered in the previous serving PCX (PCXp), the HLR sends the REGCANC operation (b) to the previous PCX (PCXp) through the VLRp. The PCXp then forwards the regcanc operation (c) to the HLR. The HLR may send the SERVREQ operation (d) requesting the SCP to execute the relevant service logic. The Service Data Management (SDM) SIB in the SCP replaces the data objects of the service group information in SDP through modification operations (e-f). Thereafter, the SCP returns the servreq operation (g) to the HLR to confirm the process of location update. Then the HLR returns the requested information to the PCXn through

the VLRn in the regnot operation (h). The Basic Call Unrelated Process (BCUP) SIB in the SCP shows how these activities are joined together to process a basic call unrelated associated services/features. The service logic instances are permitted to interact with basic location registration processing.

IV. Dynamic Queue Manager

The role of Dynamic Queue Manager (DQM) is to perform the following necessary functions for the service enhancement due to dynamic queue management [15].

- After choosing a group in accordance with a call attempt from a service user, the DQM controls the group information, subscriber's terminal status (e.g. idle, busy, inactive, etc.) and queue resources. It monitors whether an idle subscriber's terminal exists or not, and dynamically allocate queue size. With this information, then it determines if the call should be stored in queue or a service subscriber's terminal can be chosen.

wireless terminal, it might change the group information according to the subscriber's terminal mobility.

The Service Logic Execution Manager (SLEM) generates, activates and deletes the Service Logic Program Instance (SLPI) as well as supports necessary functions during the SLPI operation. Among its components, the Service Logic Selection/Interaction Management (SLSIM) generates an appropriate SLPI, and supports the interactions between SLPIs and between SLPI and other components (external functional entities and internal components to SCP).

A SLPI is generated and activated for each IN call, and the necessary actions are performed to provide the corresponding service. A various structures for SLPI are proposed according to their ways of implementation that are used according to each commercial SCP. It is assumed in this paper that the structure of SLPI is such that the information of GSL (fundamental process logic of SLPI), and SIB classes for GSL are included, and each SIB is processed with respect to the information of GSL. Therefore, Figure 4 is represented such that SLPI has GSL information (GSL Info.), and SIB classes for GSL.

Resource Manager plays a role of providing the necessary resource allocation/release functions when the SLPI is requesting the internal resource of SCP during its running. A various resource managers can exist within Resource Manager, and the DQM proposed in the paper is one of the resource managers. To enable the interaction between the DQM and the SLPI, the related SIBs (Queue SIB, Distribution SIB) internally have the functions for the interaction, through which SLPI can pass data to the DQM and vice versa.

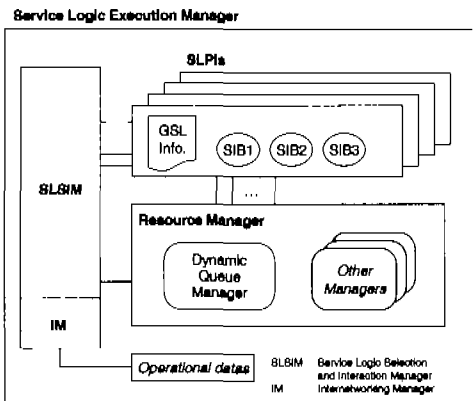


Fig. 4 Service logic execution manager structure

- When a call has been completed, it releases the resources (i.e., assigns the status of service subscriber's terminal idle), and it allocates resources to the call if there are calls waiting in the queue for the group.
- When the service subscriber's terminal is the

4.1 Internal service logic scenario

The information flows providing the mass call based IN service (e.g., Mass Calling, Televoting, Freephone service, Universal Access Number etc.) are depicted in Figure 5 through Figure 7. In these figures, the information flows can be

divided into, in a large sense, the procedure for the call service request (Figure 5, Figure 6) and the procedure for the location registration due to the subscriber's roaming (Figure 7). This section describes the internal procedure scenario of SCP for each case. Basically, this section only deals with the interaction between the SLPI and the DQM except the interactions among the SCP, SSP, IP and SDP.

The following cases are that SLPIs interact with the DQM for the IN service request.

- (1) At the request of the calling party, the DQM whether or not the service can be provided. That is, the DQM examines the status of resources (i.e., called parties).
- (2) In case the call is queued because no idle called party exists
- (3) In case no idle called party exists and there is no space in queue, therefore the call has to be terminated.
- (4) In case an idle called party exists, so the call can be directly connected.
- (5) In case a calling party waiting in queue abandons the call
- (6) In case call can be connected from queue because one of called parties has just become available.

Each case is dependent of each other, and they can be classified as follows:

- (a) (1)→(2)→(6): The call processing is continued after queuing.
- (b) (1)→(4): The call is directly processed without queuing.
- (c) (1)→(2)→(5): The call is queued, however the call is terminated by some reason (by a user's cancellation or by the expired timer).
- (d) (1)→(3): The call has to be terminated without queuing (i.e., queue full).

The interaction scenario between the SLPI and the DQM for the case (a) and (b) is shown in Figure 5. Queue Monitor SIB operation will be succeeded by user interaction session close SIB operation.

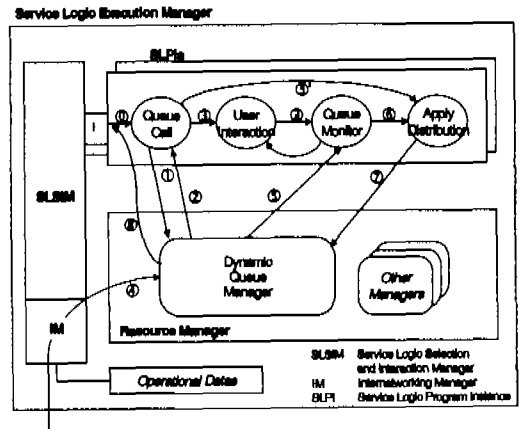


Fig. 5 Interaction between SLPI and DQM (1)

During the process of providing the IN service with queue service feature, the SLPI firstly activates Translate SIB and Log SIB (0), then run Queue Call SIB to find an idle called party. Thereafter, the rest of the scenario is as follows:

- (1) To find an available called party to be connected to the calling party, Queue Call SIB calls the DQM.
- (2) After searching the corresponding group information from the internal table, the result is returned to Queue Call SIB. The result message returned tells either the call is queued because there is no idle called party, or the call can be directly continued because there is an idle called party.
- (3) This is when the call is queued. The call is waiting in Queue Monitor SIB until there is an idle called party to be connected, and User Interaction SIB is simultaneously executed to announce the waiting message to the calling party.
- (3) This is when the call is not queued. In order to choose one of idle called parties, Apply Distribution SIB is executed. Thereafter, (7) (8) are processed.
- (4) When the call is queued, the DQM is reported from SSP that an idle called party has just become available through CurrentStatusReport (idle) notification.

- (5) The DQM reports Queue Monitor SIB in SLPI that serves the calling party waiting for an idle called party that an idle called party has become available.
- (6) Queue Monitor SIB makes Apply Distribution SIB executed.
- (7) The DQM received distribution algorithm and parameters from Apply Distribution SIB.
- (8) The DQM executes an appropriate distribution algorithm to select a suitable called party. Using the called party selected by the DQM, according to the predefined procedure, the calling party is connected to the called party, and the call is continued.

following describes the rest procedures in Figure 6 where case (c) and (d) are included.

- (1) To find an available called party to be connected to the calling party, Queue Call SIB calls the DQM.
- (2) After searching the corresponding group information from the internal table, the result is returned to Queue Call SIB. The result message returned tells either the call is queued because there is no idle called party, or the call can be directly continued because there is an idle called party.
- (3) If the call is queued, User Interaction SIB is executed to announce the waiting message. Then Queue Monitor SIB is executed.
- (3) 'This is the case when there is no idle called party and there is no space in queue. The call termination procedure is carried out.
- (4) The DQM reports that the queue timer has expired for the corresponding call, and the call is removed from the queue. Queue Monitor SIB then sends an announcement telling the call is to be terminated, and the logic jumps to the logical end, queue timer expiry, to finish the call. (6)
- (4) 'The message telling the call is abandoned (EventReportBCSM) is passed from SSP. In this case, step (5) is processed.
- (5) Queue Monitor SIB reports that the calling party has canceled the call to enable the DQM to release the corresponding resource for the call. Thereafter, the logic jumps to the logical end, calling party abandon.
- (6) The call terminating process is performed. If this is right after step (4), the announcement telling that the call is to be terminated due to the timer expired is sent to the calling party before the call termination. If this is right after step (5), the call information is logged before the call termination.

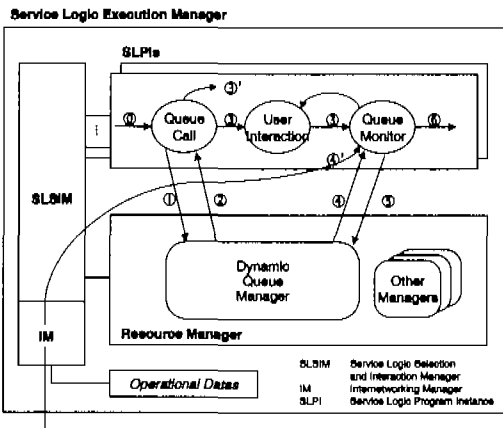


Fig. 6 Interaction between SLPI and DQM (2)

The messages passed from the DQM to Queue Call SIB are not only those in step (2) but also that telling the queue has no more allocable resource. Moreover, the return value of Queue Monitor contains the messages telling the queue timer has expired and telling the caller has abandoned the call, besides that in step (5) telling an idle called party has become available. Those messages are used in the case (c) and (d), and the interaction scenario between the SLPI and the DQM in those cases are depicted in Figure 6.

During the process of providing the IN service with queue service feature, SLPI firstly run Translate SIB and Log SIB (0), then run Queue Call SIB to find an idle called party. The

When a IN service subscriber moves, the group information to which the subscriber belongs should be updated according to the new location of the subscriber. To process this, when the

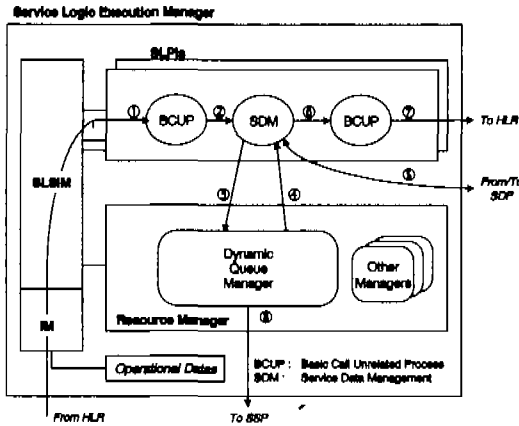


Fig. 7 Interaction between SLPI and DQM (3)

change of the location is received from the HLR, the SCP interacts with the DQM to update the group information. The update scenario performed in the SCP according to the movement of the subscriber is shown in Figure 7.

The following describe the location registration procedure for a subscriber who has just moved to other location groups.

- (1) The HLR reports the location update information of the subscriber to the SCP (SERVREQ), so that the SCP passes the information to the BCUP SIB in the SLPI.
- (2) The BCUP SIB analyzes the information, and the result is passed to Service Data Management (SDM) SIB.
- (3) SDM SIB sends the location update information of the subscriber to the DQM
- (4) The DQM updates the group information that the change in the location of the subscriber affects, and the result is passed to SDM SIB.
- (5) Using the received information, SDM SIB stores the corresponding group information to the SDP.
- (6) After the group information update is finished, the BCUP is executed.
- (7) The BCUP notifies the HLR that the group information update has finished (servreq), so that HLR can finish the location registration procedure.

(8) The DQM send a EveryStatusChange message to the SSF to assign an arming as TDP-N.

The two BCUPs used in the Figure 7 are not different instances, but they are divided in order to describe SIBs that are executed in order. Furthermore, In the interaction with external functional entities. The initial start message is sent through the SLSIM, and others in the middle of service are transferred directly through functional entity access manager.

4.2 DQM data structure

Figure 8 describes the data structure of the DQM which performs queue status management, group information management, and the maintenance of information of service subscriber's terminal (i.e., telephone status).

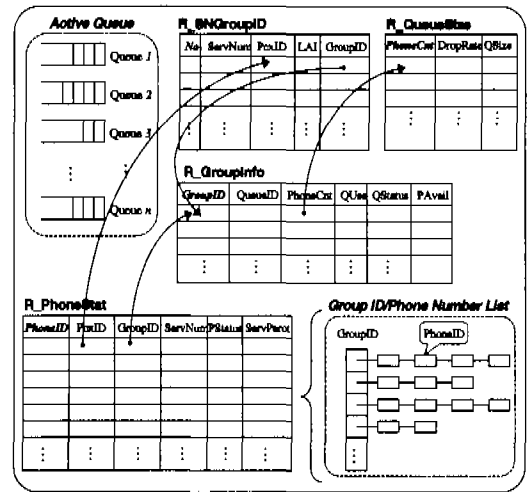


Fig. 8 Data structure of the DQM

The data structures of the DQM as shown in Figure 8 are as follows.

- Active Queue: queue resources generated for the corresponding groups.
- Service Number/GroupID Relation Table (R_SNGroupID): this table includes PCX ID (PexID), Group ID (GroupID) and the allocated service numbers designated by service providers (ServNum). The number of PCXs in one group can be one or more.

- Group Information Relation Table (R_Group-Info): this table includes the number of wired and wireless terminals in the group (PhoneCnt), active Queue ID (QueueID), Queue Usage (QUse), Queue Status (QStatus) and the number of available terminals in the corresponding group (PAvail).
- Phone Status Relation Table (R_PhoneStat): this table includes, for each subscriber's wired and wireless terminal (PhoneID), the ID of the PCX that the terminal belongs to (PcxID), enrolled group (GroupID), Service Number of SAC (ServNum), Phone (i.e., terminal) Status (PStatus), and Service distribution Percentage so far (ServPerct).
- Queue Size Relation Table (R_QueueSize): with on-line statistical methods or on-demand request from the manager, it maintains queue sizes (QSize) in proportion to the number of terminals (PhoneCnt) in a group, and the service drop rate (DropRate) which depends on the grade of service policy of service provider.
- GroupID/Phone Number List: it is internally used to perform a fast search for an available terminal in the relevant group. At the initialization time, DQM creates the linked list on memory by considering the relationship between the Phone Status Table and Group Information Table.

4.3 Management of queue size

Queue Size Table stores the sizes of queues and the service drop rates according to the number of terminals in the group. The DQM determines the values of records in Queue Size Table by using the queuing theory techniques to calculate appropriate values. As shown in Figure 9, we assume a parallel service server model, in which there is a limit on the size of queue in the system.

This queuing model is referred to as M/M/c/K[2], and has the following characteristics:

- arrival rate (λ) : arrival rate of call origination per unit time.

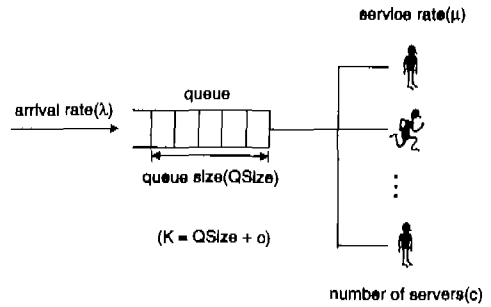


Fig. 9 M/M/c/K model

- service rate (μ) : service rate per unit time.
- number of servers (c) : the number of service subscriber's terminals per group.
- system size (K) : the number of service subscriber's terminals plus queue size.

In the M/M/c/K model, the probability when the system is empty (P_0) yields (Eq. 1) and the probability of n calls in the system (P_n) yields (Eq. 2).

$$P_0 = \begin{cases} \left[\sum_{n=0}^{c-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \frac{(\lambda/\mu)^c}{c!} \frac{1 - (\lambda/c\mu)^{K-c+1}}{1 - (\lambda/c\mu)} \right]^{-1} & (\lambda/c\mu \neq 1) \\ \left[\sum_{n=0}^{c-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \frac{(\lambda/\mu)^c}{c!} (K-c+1) \right]^{-1} & (\lambda/c\mu = 1) \end{cases} \dots\dots(\text{eq. 1})$$

$$P_n = \begin{cases} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n P_0 & (0 \leq n < c) \\ \frac{1}{c^{n-c} c!} \left(\frac{\lambda}{\mu}\right)^n P_0 & (c \leq n \leq K) \end{cases} \dots\dots(\text{eq. 2})$$

P_0 and P_n are used to obtain the average number of calls in the queue (L_q). We calculate L_q by using (Eq. 3).

$$L_q = \frac{P_0 (c/\mu)^c}{c!(1-\rho)} \{1 - [(K-c)(1-\rho) + 1]\rho^c\} \quad (\rho = r/c = \lambda/c\mu \neq 1) \dots\dots(\text{eq. 3})$$

Hence, P_K (i.e., the probability of K calls in the system) and L_q are used to obtain the average waiting time in queue (W_q), and the average call

drop rate (C_d). We calculate W_q and C_d by using (Eq. 4) and (Eq. 5), where λ' is the mean rate of users actually entering the system.

$$W_q = \frac{L_q}{\lambda'} \quad \lambda' = \lambda(1 - p_K) \dots\dots\dots(\text{eq. 4})$$

$$C_{d,K} = \lambda' R_K \dots\dots\dots(\text{eq. 5})$$

The service provider can create a reasonable Queue Size Table according to the number of service terminals in the group by taking C_d into account. Once C_d (DropRate) is designated by the service provider, the DQM determines the size of queue (QSize) as the number of service terminals (PhoneCnt), which can be changed dynamically in the wireless intelligent network environment.

V. Simulation and Analysis

In this section, the appropriate size of the queue used by the DQM designed in this paper is determined, and the influence of the change in each drop rate on the size of the queue is verified. We simulate the dynamic queue management model applied according to the increment of IN service request rate and the mobility of subscriber's terminals. We select the Freephone service which is one of the commercially most interesting IN services as the evaluation model to be simulated in wired and wireless network environment.

5.1 Assumptions

In order to evaluate the DQM designed in this paper, we make an assumption as follows:

It is assumed in this paper that Freephone service is the only service that affects the allocation of the queue resource. Although the subscribers to Freephone service could belong to different service providers, it is also assumed here that the service is provided for one service provider. Although one group could consist of several PCXs, only one PCX is assumed for one group under the following simulation environment.

Moreover, the delay time for a call delivery (T_{Delay}) from a user to a subscriber is the total sum for user interaction time (T_{UI}), the signaling time between physical entities (T_{PE}), the SCP processing time (T_{SCP}) and the location trace time for a wireless subscriber (T_{LR}) as follows. Where T_{LT} is meaningful only if a subscriber's terminal is a mobile terminal. (Refer to Figure 2)

$$T_{\text{Delay}} = T_{\text{UI}} + T_{\text{PE}} + T_{\text{SCP}} + T_{\text{LT}}$$

$$T_{\text{PE}} = 7T_{\text{PCX_SCP}} + 2T_{\text{SCP_SDP}} + 2T_{\text{PCX_PCX}} + 3T_{\text{PCX_IP}} + 3T_{\text{SCP_IP}}$$

$$T_{\text{LT}} = 2T_{\text{PCX_HLR}} + 2T_{\text{HLR_VLR}} + 2T_{\text{VLR_PCX}}$$

Here, T_{UI} is the waiting time in the relevant queue with playing announcement to the queued call when there is no available subscriber's terminals. For this reason, it can be considered that T_{UI} is much bigger than others. Therefore, the call delay time that affects the user is assumed to be only T_{UI} .

The basic environment parameters used in the simulation for the Freephone service based on the maximum capacity of a TDX-10 PCX are as follows.

- number of users per PCX (N_{mba}) : 350,000 users
- average number of the call attempts per user (λ_{mba}) : 1.6 calls/hour
- average holding time per call (T_{service}) : 90 sec
- proportion of IN call to all normal call attempts (P_{IN}) : 5.8%

In this simulation, the proportion of Freephone service to all IN services (i.e., P_{FPH}) is assumed to be 10%.

Thus, the Freephone service call attempt rate (λ_{FPH}) becomes as follows.

$$\lambda_{\text{FPH}} = N_{\text{mba}} * P_{\text{IN}} * \lambda_{\text{mba}} * P_{\text{FPH}} / 3600 = 0.902 / \text{sec.}$$

The number of calls to a service subscriber (i.e., λ_{service} , a specific service number) is assumed to be on the increase at a 1% ~ 10% (i.e., "service request" shown in the following figures) in the above rate (i.e., λ_{FPH}).

$$\lambda_{\text{service}} = \lambda_{\text{FPH}} * \text{service request}$$

The increment of the call attempt rate according to the increment of the IN service request rate can be as the increment of the call attempt rate for a subscriber in the simulation.

5.2 Simulation results

From the simulation based on the above parameters, the queue size can be verified according to the number of subscriber's terminals in proportion to the rate of increase on specific service request. The appropriate waiting time for each case is calculated, and the appropriate queue size required according to the call attempting rate is obtained.

Figure 10 shows the queue size required according to the each number of subscriber's terminals (i.e., servers shown in the following figures) for the drop rate of 1%. We observe that as the service request rate increases, the required queue size increases exponentially. With the periodic statistics data of the service users distribution, the optimization of the queue resource maintained in the system can be achieved when the DQM in this paper is applied.

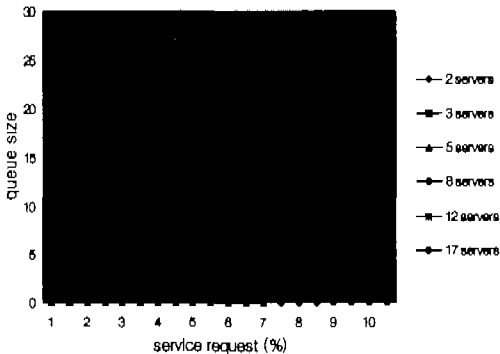


Fig. 10 Queue Size according to the number of servers in the case of 1% drop rate

Figure 11 shows the queue size required to maintain the drop rate of 0.1%, 1% and 10% when the number of the subscriber's terminals is fixed to eight. When the specific service rate is the same, the changes in the drop rate affect largely the queue size. Thus, when the allocation of the queue resource is difficult, the method of

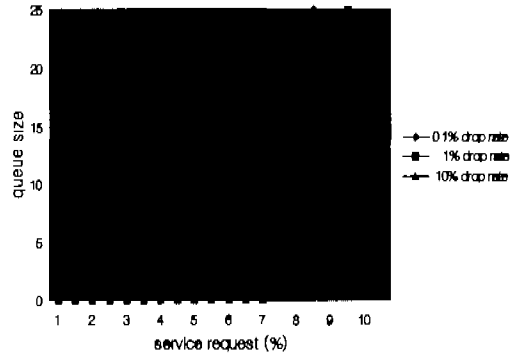


Fig. 11 Queue Size according to the drop rate in the case of 8 servers

decreasing the appropriate queue size through the change in the drop rate can be used.

Figure 12 shows the service user waiting time in the queue for the case of Figure 10. Only the calculated waiting time in the queue is considered. The service user waiting time in the queue is apt to change steeply as it was to change in the queue size. However, the degree of change becomes smaller as the number of the subscriber's terminals increases. For instance, if the users' characteristics of waiting time is assumed to be less than 30 seconds, and the service request is more than three percentages, the service attempt can be canceled when the number of the subscriber's terminals is two or three. Yet, if the number of the subscriber's terminals is more than that, all user calls waiting in the queue will be connected.

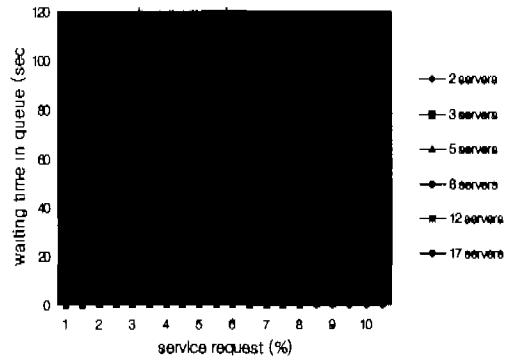


Fig. 12 Waiting time in the queue according to the number of servers in the case of 1% drop rate

Figure 13 shows the changes in the waiting time in the queue for the drop rate of 0.1%, 1% and 10% with the eight subscriber's terminals. It can be seen that the waiting time decreases steeply as the drop rate increases.

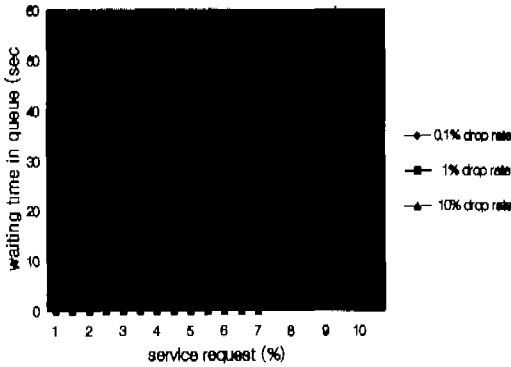


Fig. 13 Waiting time in the queue according to the drop rate in the case of 8 servers

Figure 14 shows the necessary queue sizes to guarantee all the accepted calls, which are chosen within 1% drop rate, wait no longer than 30 seconds in the queue on average. Comparing with Figure 10, where the waiting time in queue is unlimited, only moderate service requests can be served to satisfy both 99% call completion rate and maximum average 30 seconds in queue. It is obvious that, in order to meet 99% call completion rate in case of heavy service request, a much bigger queue size should be allocated, which would consequently lead longer average waiting time than 30 seconds in queue.

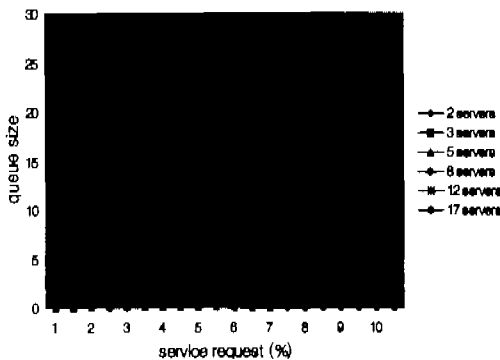


Fig. 14 Queue size according to 30 max. 30 seconds waiting time in the queue in the case of 1% drop rate

VI. Conclusions

Fast provisioning of new services to end users has rapidly become the key issue in the modern telecommunications environment, often being the most important element determining the success of operators and manufacturers considering both wired and wireless network scenarios. Network convergence is taking place at many levels. From the telephony viewpoint, the current trend is to provide common services available in both wired and wireless domains. In this context, the IN provides a key concept, since it represents today the globally accepted service platform for integrated service creation and provision. From the viewpoint of call service completion, this convergence concentrated mass calls for a unified handling of services during a certain time, involving wireless IN subscribers as well as wired ones, may be especially considered to be important. This paper presents a queue service feature for the service enhancement to cope with the mass calling situation. It describes the design and implement of a DQM with dynamic management mechanism that considers not only wired subscribers but also wireless subscribers reflecting their mobility. The DQM can dynamically control the information of the group, which a subscriber's terminal belongs to, according to the subscriber's terminal mobility and the change of the call status. The size of queues for each group can be configured by the call drop rate that the subscriber demands. In order to apply the queue size assignable by the designed DQM, a simulation based on the number of the subscriber's terminals and the drop rate is performed by taking the service increment rate and the mobility of subscriber's terminals into account. In addition, the appropriate waiting time in the queue as required is simulated according to the number of subscriber's terminals. Moreover, we design and implement the internal service logic interacting with the corresponding SIBs and its data structure in the DQM. The dynamic

