國立交通大學

資訊科學與工程研究所

博 士 論 文

行動電信網路中即時計費之研究

A Study for Online Charging in Mobile Telecommunications

研 究 生：李欣怡

指導教授：林一平　教授

中 華 民 國 九 十 九 年 八 月

行動電信網路中即時計費之研究

A Study for Online Charging in Mobile Telecommunications

研 究 生：李欣怡　　Student：Hsin-Yi Lee

指導教授：林一平 博士　　Advisor：Dr. Yi-Bing Lin

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

博 士 論 文

A Dissertation

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

August 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年八月

行動通訊網路中即時計費之研究

學生：李欣怡　　指導教授：林一平 博士

國立交通大學資訊科學與工程研究所博士班

摘　要

全球行動通訊系統(UMTS)為第三代行動通訊(3G)的主流規格之一。第三代行動通訊規格組織(3GPP)第五版中提出了 IP 多媒體子系統(IMS)，以提供多媒體服務。若要成功推廣 IMS 服務，如何正確及合理地收取封包費用成為行動電信業者關心的議題。建置 IMS 服務需要精確的計費管理系統以及有效率的服務提供機制，因此，3GPP 第五及第六版提出了一個基於 IP 的即時計費系統(OCS)。透過即時計費機制，電信業者可以更彈性地針對每筆交易處理帳戶餘額以及網路資源授權等。

3GPP 第八版提出了長期演進技術(LTE)，此技術為 UMTS 的演進。由於 LTE 提供更高的效能以及更低的延遲時間，服務品質(QoS)控制成為有效資源管理的重點項目之一。QoS 控制的要求包括：在不同 QoS 等級中的使用者必須能夠以不同的費率計費，且電信業者必須能夠處理資料傳輸並提供合適的 QoS 給使用者。QoS 控制以及彈性計費的議題都在策略與計費控制系統(PCC)中提及。根據事先定義的 PCC 規則，UMTS/LTE 管控 IMS 應用服務使用之 IP 網路資源(如預留之頻寬)。在 PCC 機制中，計費決策受到許多因素影響，其中包含：服務型態、使用量、以及提供的 QoS 等。

本論文探討即時計費及策略控制之議題。在本篇論文的第一部份，我們首先研究即時計費中 OCS 的點數保留機制。OCS 之設計需要降低信號控制訊息的交換，以及增加系統處理計費的效能。若分配給某應用服務之點數在服務結束之前消耗

完畢，則此服務需要再向 OCS 請求更多點數。在請求點數的同時，封包傳遞會被暫停，直到此服務向 OCS 取得點數為止。為了避免服務之暫停，我們提出了點數提前保留機制，也就是在點數耗盡之前先向 OCS 發出點數請求。我們發展出數學模型和模擬實驗，以準確分析影響系統效能之各項指標。根據本研究的結果，我們提出的方法能為行動業者提高即時收費系統的效能。

本論文的第二部份中，我們設計並實做 LTE 中的 PCC 系統。本系統根據應用服務中的資訊來做 PCC 決策，而這些 PCC 決策規則可以由使用者定義。當一個服務請求符合某個決策規則時，此 PCC 系統會根據 QoS 資訊向 OCS 提出計費請求。

在論文的第三部份，我們開發了基於 PCC 系統創建新應用服務的服務平台。藉由本平台提供的標準的網路服務(Web Service)應用程式介面(API)，應用程式開發者可以在不需知道電信網路的細節(如 PCC 協定)的情況下創建新的應用服務。我們以群組計費系統(GAS)應用服務為例，描述在本服務平台上如何輕易的創建新的計費應用服務。

本論文提出的研究成果，可以在電信網路中作為未來研究即時計費及策略控制之重要參考依據。

關鍵字：IP 多媒體子系統, 全球行動通訊系統, 策略控制, 即時計費

A Study for Online Charging in Mobile Telecommunications

Student: Hsin-Yi Lee Advisor: Dr. Yi-Bing Lin

Institute of Computer Science and Engineering

National Chiao Tung University

Abstract

Universal Mobile Telecommunications System (UMTS) is one of the major standards for the third generation (3G) mobile telecommunications. The 3G Partnership Project (3GPP) Release 5 introduced the *IP Multimedia Subsystem* (IMS) to provide multimedia services.

In order to successfully promote IMS services, how to charge packet data service accurately and reasonably has become a major concern of operators. The deployment of the IMS services requires effective charging management system and efficient service delivery mechanism. Therefore, the 3GPP Release 5 and 6 proposed the IP-based *Online Charging System* (OCS) to incorporate data applications with real-time control and management. Through online charging, an operator can ensure that credit limits are enforced and resources are authorized on a per-transaction basis.

The 3GPP Release 8 introduced *Long Term Evolution* (LTE) that is a set of enhancements to UMTS. Since LTE offers higher throughput and lower latency, *Quality of Service* (QoS) control becomes an important issue for cost-effective resource management. It is desirable that the subscribers who require different QoS levels should be charged with different rates, and telecom operators negotiate the data transfer and offer appropriate QoS for the subscribers. The QoS control and flexible charging issues are addressed in the *Policy and Charging Control* (PCC).

Based on the predefined PCC rules, the UMTS/LTE manages and controls the IP network resources (e.g., the allocated bandwidth) to the services. The factor that affects charging decision includes the service type, the amount of usage and the provisioned QoS.

In the first part of the dissertation, we study the online credit reservation in the OCS. Specifically, the design of OCS needs to reduce the signaling overhead and improve the system performance. If the assigned credit units are consumed before the session is completed, the session needs to request more credit units from the OCS. During the credit request operation, packet delivery is suspended until extra credit units are granted from the OCS. To avoid session suspension, we propose the credit pre-reservation mechanism that reserves credit earlier before the credit is actually depleted. Analysis and simulation experiments are conducted to investigate the effects of input parameters. As a result of our study, the mobile operator can achieve high performance in the online charging management.

Second, we investigate the design and the implementation of the PCC system in LTE. According to the session information, the PCC makes policy decisions, where the policy rules can be formulated based on user-defined information. When a service request meets a policy rule, our PCC system initiates a charging operation toward the OCS based on the QoS information.

In the third part, we develop an IMS service platform for new service creation based on the PCC system. Through the standard-based web service APIs furnished by the service platform, application developers can create services without knowing the details of the telecommunication network such as PCC protocols. We use Group Accounting System (GAS) as an example to illustrate how a new charging application can be easily created and provided in our service platform.

These research results presented in this dissertation can be viewed as a useful foundation

for further studies in policy and charging in mobile telecommunications.

Keywords: IP Multimedia Subsystem (IMS), Universal Mobile Telecommunications System (UMTS), policy control, online charging

Acknowledgements

I am deeply indebted to my advisor Prof. Yi-Bing Lin for his continuous support, encouragement, and guidance throughout my research. His extensive knowledge and creative thinking have been an invaluable help for me. Lin taught me how to approach a research problem and the need to be persistent to accomplish any goal. He was always there to meet and give advice. Without his supervision, I would not have completed this dissertation.

I would like to gratefully and sincerely thank my committee members, Prof. Ming-Feng Chang, Prof. Chein-Chao Tseng, Prof. Han-Chieh Chao, Prof. Jean-Lien Chen, Dr. Sheng-Lin Chou and Dr. Herman Chung-Hwa Rao who gave insightful comments and reviewed my work on a very short notice.

I also express my appreciation to all the faculty, staff and colleagues in the Department of Computer Science. In particular, I would like to thank Prof. Yu-Chee Tseng, Prof. Rong-Jaye Chen, Prof. Sok-Ian Sou, Prof. Phone Lin, Dr. Shih-Feng Hsu, Dr. Meng-Hsun Tsai, Dr. Yung-Chun Lin, Dr. Ya-Chin Sung, Chien-Chun Hung-Fu, Pin-Jen Lin, Ren-Huang Liou, Shiou-Wen Chu, Zheng-Han Wu, Samuel Sung, Jenny Liang, Rainee Yeh and the other labmates in Laboratory 117.

I would like to thank Rebecca Chen, Li-Fen Li, Webbor Lee and Yi-Hong Wang for their friendship and help during my internship in IBM.

Let me say “thank you” to my best friends Pei-Hua Chen, Jeany Ling, Doris Chen, Cafemilk Hsieh, Yu-Ying Huang, Christina Lin, Bie Chen and Becky Chen for their friendship and supports in various ways. I would like to extend my heartfelt gratitude to Kostiantyn Samoilenko, Phanix Chen, Edward Kao, Yu-Shiang Chang, Jau-Yi Wang, Corey Lee Bell and Ruwan Indika. Their positive attitude encourages me especially in times of difficulty and frustration.

Special thanks go to Federico Agustin Altolaguirre and Hanjo Lu, who have confidence in me and dedicate their precious time for the presentation of my studies.

Last but not the least, I am grateful to my dear parents, Shih-Chao Lee and Mei-Yueh Chen, for unfailing love and firmly support in my life; to my sister Chia-Chueh Lee, for the encouragement to pursue my interests.

This work was supported by the IBM Ph.D. Fellowship and Pat Selinger Ph.D. Fellowship.

Sharon, Hsin-Yi Lee

2010

Contents

Abstract in Chinese **i**

Abstract in English **iii**

Acknowledgements **vi**

Contents **viii**

List of Tables **xi**

List of Figures **xii**

1 Introduction **1**

1.1 The UMTS/IMS Network 3

1.2 Online Charging System 6

1.2.1 Diameter 8

1.2.2 The Diameter Message Flow 9

1.3 Policy and Charging Control 12

1.4 Organization of the Dissertation 14

2 Credit Pre-reservation Mechanism for UMTS Prepaid Service 16

2.1 Credit Pre-reservation Mechanism 17

2.2 Analytic Model for the CPM 20

2.3 Numerical Examples 25

2.4 Conclusion 30

3 Policy and Charging Control System for Advanced Mobile Services 33

3.1 Policy and Charging Control Architecture in LTE Network 35

3.2 The PCC Testbed 36

3.2.1 The Functional Blocks of the PCC Testbed 37

3.2.2 The Message Flow of the PCC Testbed 42

3.3 Performance Measurement 44

3.4 Conclusion 46

4 Transparent Charging for IMS Services 50

4.1 Parlay 51

4.2 IBM WebSphere software for Telecom 52

4.3 Service Provision for WsT-Based Group Accounting System 55

4.4 Message Flows for Group Accounting System 57

4.5 Application Development 60

4.6 Conclusion 62

5 Conclusions and Future Work 64

5.1 Concluding Remarks 64

5.2 Future Work 65

A The Simulation Model for Credit Pre-reservation Mechanism **75**

B The Implementation of the PCC Diameter Modules **83**

List of Tables

1.1 Diameter Command Codes 8

3.1 The Message Data Signaling Delay Measured in Our Testbed 46

4.1 A Table Entry in the OCS's Account Database 56

4.2 The GAS Entry in the CSCF Routing Table 56

List of Figures

1.1 The UMTS/IMS Network Architecture 4

1.2 The Online Charging System Architecture 7

1.3 Message Flow of the Diameter Credit Control Mechanism 10

1.4 PCC architecture for IMS service 12

2.1 Flowchart of the CPM (Steps C-1 and C-6 refer to Steps D-1, D-2, D-6 and D-7 in Figure 1.3 18

2.2 Validation of simulation and analytic results on P_r and B ($\alpha = 0.01$, $\gamma/\mu = 1/20$, and $\delta = 0.3\theta$; Solid curves: analytic results; dashed curves: simulation results) 24

2.3 Effects of λ/μ and C on P_{nc} and X_s (α=0.01,γ/μ=1/20, δ=0.3θ, and θ=50λ) . . 25

2.4 Effects of θ and the packet interarrival time distribution (α=0.01, γ/μ=1/20, δ=0.3θ, $C = 600/\alpha$, and λ/μ=3) 28

2.5 Effects of δ and the RU operation delay distribution (α=0.01, γ/μ=1/20, θ=100λ, $C = 600/\alpha$, λ/μ=3, and the packet arrival times have the Pareto distribution with the mean $1/\lambda$ and $b = 2$) 31

3.1 Policy and Charging Control Architecture in LTE Network 34

3.2 The Block Diagram for the PCC Testbed Implementation 36

3.3 Call Duration Control Program Segment 39
3.4 Bandwidth Control Program Segment 39
3.5 The Message Flow for the IMS Call Control Service in PCC 41
3.6 A Snapshot for the CCR Message Handling in Our Testbed 46
3.7 A Snapshot for the CCA Message Handling in Our Testbed 47
3.8 The List of Diameter Messages Captured in the PCRF 48

4.1 The Parlay Architecture 51
4.2 IBM WebSphere software for Telecom 53
4.3 The Graphical User Interface of iPhone for Group Accounting System 55
4.4 Message Flows for Group Accounting System 58
4.5 StartCallDirectionNotificationRequest Web Service Program Segment 61

A.1 Flowchart of the simulation model for the CPM 77
A.2 Flowchart of the CCR module 80
A.3 Flowchart of the CCA module 81

B.1 Class Diagram of Our Purposed Open Diameter Application 84

Chapter 1

Introduction

The *second generation* (2G) technology has provided cellular network services to over three billion people worldwide. These 2G services include telephony calls and text messages (e.g., *Short Message Service*; SMS). Charging in the traditional 2G telephony is comparatively simple because there is typically only one voice call session at a time. During the call setup/release process, the cellular network records the call-related information and then uses this information for rating and billing.

The 2G networks have evolved to the *Third generation* (3G) for advanced mobile telecommunications. The 3G aims to integrate two of the most successful technologies in communications: cellular networks and the Internet. The Internet environment encourages global usage with flat-rate tariffs and low entry costs. A major problem of the "flat-rate" tariffs is that such a business model cannot justify the expensive equipment/operation investments of mobile services. Mobile telecom operators have to move from a bit-pipe model to a revenue-generating services model. To integrate IP with wireless technologies with the "right" business model, the *Third Generation Partnership Project* (3GPP) has specified *Universal Mobile Telecommunications System* (UMTS) all-IP architecture to enable web-like services and a new billing paradigm

in the telephony world [35]. In UMTS, the core network consists of two service domains: the circuit-switched (CS) and the packet-switched (PS). The 3GPP Release 5 introduces the *IP Multimedia Subsystem* (IMS) on top of the PS service domain to enable mobile data services [20]. Unlike the 2G circuit-switched call, multiple prepaid sessions can be accommodated simultaneously in the packet-switched environment. Thus, the traditional charging mechanism can not be applied. In other words, this evolution requires new mechanisms to collect information about chargeable events and to impose flexible mobile billing schemes (such as time-based, volume-based or content-based) [34, 32, 4, 6].

A telecom operator typically provides offline charging where the users pay for their services periodically (e.g., at the end of a month). On the other hand, online charging is used for prepaid services, which means the user has to make an advanced payment before the service is delivered. By merging the prepaid and postpaid methods, the *Online Charging Systems* (OCS) [17, 14] has been proposed in The 3GPP Release 5 and 6 to allow both offline and online services to be charged in real-time. The real-time solution provides two-way communication between network nodes and the charging/billing system, which transfers information about rating, billing and accounting. Through online charging, the operator can ensure that credit limits are enforced and avoid bad loan. From a subscriber's perspective, knowing the charges in advance and having self-imposed credit limits can make himself control the budget.

A major charging issue in packet-switched domain is that the Internet typically provides a best-effort service without Quality of Service (QoS); that is, the network does not guarantee the minimum amount of bandwidth for a particular connection or the maximum delay of the transmission. Whereas the network resources consumption may vary dramatically among subscribers for Internet access, it is necessary for telecom operators to provide and control the QoS. In this case, subscribers with different QoS requirements should be charged by different

rates. The 3GPP Release 7 has defined the *Policy and Charging Control* (PCC) [21], which includes the QoS control and charging control in UMTS/IMS. According to the session and media-related information, the PCC makes policy decisions, where the policy rules can be formulated based on user-defined information (such as a subscription profile). When a service request meets a policy rule, the PCC triggers a desired action (such as accepting the service with the requested bandwidth), and initiates a charging operation towards the OCS based on the QoS information.

This chapter first presents the UMTS/IMS network architecture and its IP-based online charging system. Then we describe the policy and charging control mechanism to provide service-aware QoS management in charging. Finally, we discuss application-level charging and the organization of this dissertation.

1.1 The UMTS/IMS Network

The first deployment of the UMTS is in the Release 99 architecture. The UMTS network has evolved from *Global System for Mobile Communications* (GSM) and *General Packet Radio Service* (GPRS) [3, 22]. The 3GPP Release 5 introduces the IMS to effectively integrate mobile technology with the Internet [12, 13]. This section introduces the UMTS/IMS architecture.

As illustrated in Figure 1.1, the UMTS/IMS network architecture consists of the radio access network (RAN; Figure 1.1 (a)), the UMTS core network (Fig 1.1 (b)), and the IMS network (Figure 1.1 (c)). In this figure, the dashed lines represent signaling links and the solid lines represent data and signaling links. The IMS signaling protocols allow the telecom operators to offer attractive services to their customers. Such protocols are like *Session Initiation Protocol* (SIP) [42, 29] for session signaling and Diameter protocol for *Authentication, Authorization*

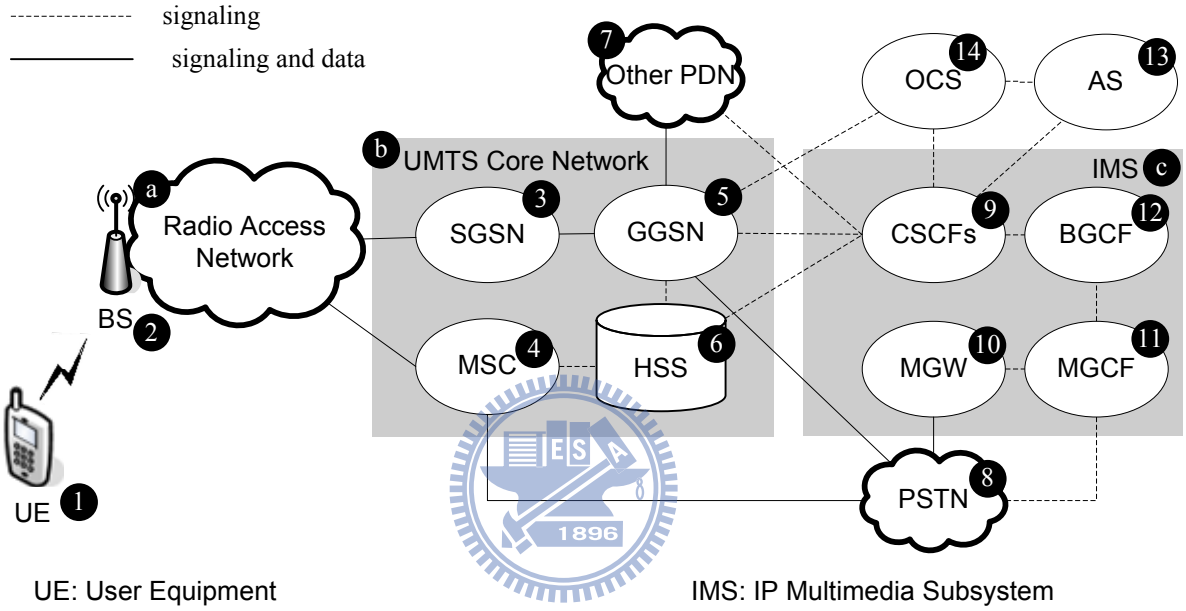


Figure 1.1: The UMTS/IMS Network Architecture

and Accounting (AAA).

In this architecture, a mobile user utilizes a *User Equipment* (UE; Figure 1.1 (1)) to communicate with the network through the RAN, or more specifically, macro or femto *Base Stations* (BSs; Figure 1.1 (2)) [33]. In the core network, the *Mobile Switching Center* (MSC; Figure 1.1 (3)) handles all the circuit-switched operations (which is about *Public Switched Telephone Network* (PSTN; Figure 1.1 (8))) while the *Serving GPRS Support Node* (SGSN; Figure 1.1 (4)) handles all the packet-switched operations and transfers all the data in the network. The SGSN connects the *Gateway GPRS Support Node* (GGSN; Figure 1.1 (5)) to access the external *Packet Data Network* (PDN; Figure 1.1 (7)) or the IMS network. The *Home Subscriber Server* (HSS; Figure 1.1 (6)) contains user-related subscription information, which can be utilized by core network nodes such as MSC, SGSN, GGSN, and IMS network.

In the IMS network, call and session control is implemented in the *Call Session Control Function* (CSCF; Figure 1.1 (9)), which acts as a SIP server to facilitate SIP session setup and teardown. The *Media Gateway* (MGW; Figure 1.1 (10)) transports the IMS user data traffic. The MGW provides user data transport between the UMTS core network and the PSTN (including media conversion bearer control and payload processing). The *Media Gateway Control Function* (MGCF; Figure 1.1 (11)) controls the media resources in the MGW. The *Breakout Gateway Control Function* (BGCF; Figure 1.1 (12)) selects the network in which the PSTN (or circuit switched domain) breakout is to occur. If the BGCF determines that a breakout is to occur in the same network, it selects an MGCF that is responsible for interworking with the PSTN. If the breakout occurs in another IMS network, the BGCF forwards the SIP request to another BGCF or an MGCF in the selected IMS network.

In Figure 1.1, an application server (AS; Figure 1.1 (13)) provides value added IP multimedia services, which reside either in the user's network or in an external third-party location.

Through standard protocols or languages (such as Java, SIP and XML), an application developer can efficiently deploy mobile applications to launch new services and therefore reduce time-to-market. In Chapter 4, we will describe an application server with charging capability. The OCS (Figure 1.1 (14)) performs online charging and collects the billing information of the IMS with the CSCFs. Details of the charging functionalities are described in the next section.

1.2 Online Charging System

This section introduces the all-IP online charging system. In online charging, a service provider can charge its customers based on the price or the tariff of the requested service and the balance in the subscriber's account in real time. Figure 1.2 illustrates the OCS architecture defined in 3GPP 32.296 [17]. The OCS supports two types of *Online Charging Functions* (OCFs), namely the *Session-Based Charging Function* (SBCF, Figure 1.2 (1)) and the *Event-Based Charging Function* (EBCF, Figure 1.2 (2)).

The SBCF is responsible for network bearer and session-based services such as voice calls, GPRS sessions or IMS sessions. The SBCF triggers the session-based charging mode, and controls SIP sessions by appearing as a *Back-to-Back User Agent* (B2BUA) that sends messages to the initiating SIP user agents. It also performs charging on non-SIP based bearer systems (e.g, GPRS and other bearer channels) using the CAP or Ro reference point (depending on the communicating network node). The EBCF is responsible for event-based services. The EBCF triggers the event-based charging mode, and controls "one-shot" events such as short message delivery, and content downloading (e.g., for ring tones or games). The SBCF and EBCF have the ability to grant or deny the network resource usage.

The *Account Balance Management Function* (ABMF; Figure 1.2 (3)) maintains user bal-

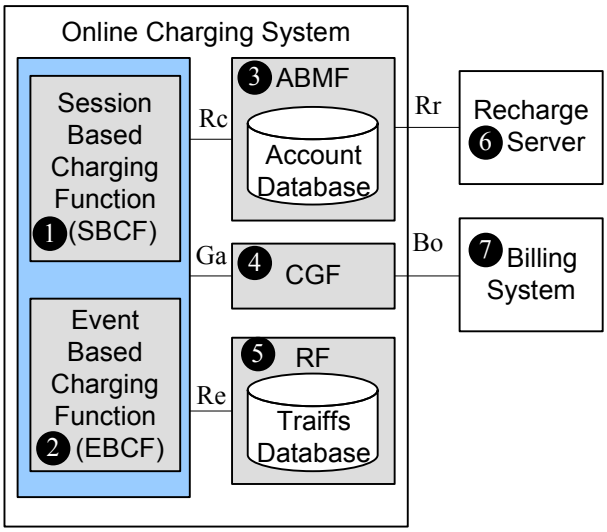


Figure 1.2: The Online Charging System Architecture

ances and other account data. When a user's credit depletes, the ABMF connects the *Recharge Server* (Figure 1.2 (6)) to trigger the recharge account function. The OCFs communicate with the ABMF to query and update the user's account. The charging data records (CDRs) generated by the charging functions are transferred to the Charging Gateway Function (CGF; Figure 1.2 (4)) in real time. The CGF acts as a gateway between the IMS/UMTS network and the billing system (Figure 1.2 (7)).

The *Rating Function* (RF; Figure 1.2 (5)) determines the price/tariff of the requested network resource (i.e, session, service or event) before and/or after service delivery. The decision about when and how to perform charging for a service session is handled by the charging policies provisioned in the RF. We note that in some cases, non-chargeable sessions (or sub-sessions) have to be explicitly monitored via "zero rating charging contexts" for consistency. The RF is responsible for providing a cost for the requested session, which can be charged by a wide variety of rateable instances such as volume, time and events. The OCF furnishes the

Table 1.1: Diameter Command Codes

Command code	Message name	Abbreviation
271	`Accounting Request/Answer`	`ACR/ACA`
258	`Re-Auth Request/Answer`	`RAR/RAA`
272	`Credit Control Request/Answer`	`CCR/CCA`

necessary information (obtained from the IMS/UMTS network nodes) to the RF and receives the rating result (monetary or non-monetary credit units).

1.2.1 Diameter

The IMS uses the Diameter protocol to transfer the accounting information. The Diameter was derived from *Remote Access Dial In User Service* (RADIUS) protocol [41] to offer more flexibility, and is generally believed to be the next generation *Authentication, Authorization, and Accounting* (AAA) protocol [24]. Diameter is an extensible protocol enabling AAA within and across IP multimedia networks. The Diameter protocol has fail-over capabilities, and it runs over, for example, secure TCP/SCTP transport. Its modular architecture offers a flexible base protocol which allows application-specific extensions. The Diameter has proven successful in overcoming the limitations of RADIUS. Therefore, rapid growth in the usage of Diameter-based charging can be expected. The 3GPP has chosen the Diameter protocol to enable IMS network AAA capabilities [17, 2]

Like the RADIUS, the Diameter follows the client-server architecture where a client and a server interact through the Diameter request and answer message exchange. Several Diameter applications defined by *Internet Engineering Task Force* (IETF) are utilized in IMS, including the Diameter Credit Control (DCC) application for IP-based online charging control [30, 15].

Each Diameter message is assigned a command code to identify its message type, which

is used for both Requests and Answers. Some examples of the command codes for Diameter charging messages are listed in Table 1.1. In this table, `Accounting Request/Answer (ACR/ACA)` commands support basic accounting, such as capacity planning, auditing, billing and cost allocation. `Re-Auth Request/Answer (RAR/RAA)` commands initiate re-authentication service for a session; for example, in a prepaid service, the Diameter server that originally authorized a session may need some confirmation that the user is still using the services. `Credit Control Request/Answer (CCR/CCA)` commands are responsible for the DCC mechanism. The mechanism includes the process of checking whether credit is available, credit reservation, deduction of credit from the end user account when service is completed and refunding of reserved credit that is not used. The message flow for DCC mechanism is explained in the following subsection.

1.2.2 The Diameter Message Flow

The Diameter message flow for session-based online charging includes three types of credit control operations: INITIAL_REQUEST (Steps D-1 and D-2 in Figure 1.3), UPDATE_REQUEST (Steps D-3 and D-4 in Figure 1.3) and TERMINATE_REQUEST (Steps D-5 and D-6 in Figure 1.3). The following operations are executed for session-based services.

Step D-1. To initiate the service session with the credit reservation, the network node (e.g., GGSN or CSCF) sends a `CCR` message with type "INITIAL_REQUEST" to the OCS. This message indicates the amount of requested credit units.

Step D-2. Through the interactions among the SBCF, the ABMF and the RF, the OCS determines the tariff of the requested service session and then reserves an equivalent amount of credit units for the service session. After the reservation is performed, the OCS acknowl-

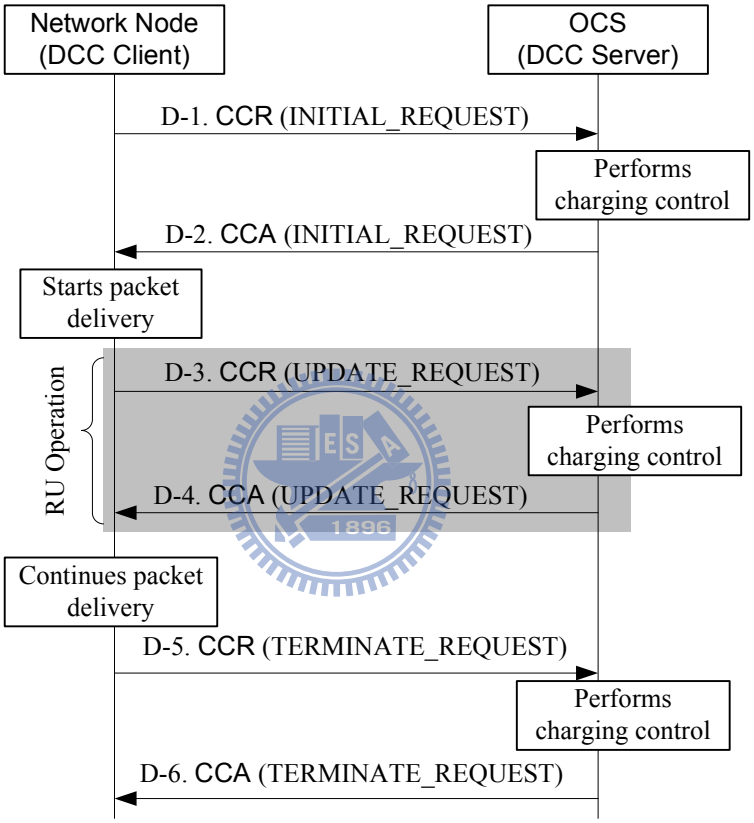


Figure 1.3: Message Flow of the Diameter Credit Control Mechanism

edges the network node by sending a CCA message with type "INITIAL_REQUEST" including credit reservation information. After receiving this message, the network node starts to deliver user packets for the session.

Step D-3. During the service session, the granted credit units may be depleted. If so, the network node sends a CCR message with type "UPDATE_REQUEST" to the OCS to report the used credit units. The network node reports the amount of used credit, and requests for additional credit units. Packet delivery is suspended, and any newly arriving data packets are buffered at the network node.

Step D-4. When the OCS receives the CCR message, it debits the amount of consumed credit and reserves extra credit units for the service session. The OCS acknowledges the network node with a CCA message including the amount of credit units that have been reserved. After the receiving this message, the network node continues the packet delivery. Note that Steps D-3 and D-4 may repeat many times before the service session is complete. At Steps D-2 and D-4, if OCS cannot afford the requested amount of credit units, the session is forced to terminate. For the discussion purpose, Steps D-3 and D-4 are called a *reserve units* (RU) operation.

Step D-5. When the session is completed, the network node sends a CCR message with type "TERMINATE_REQUEST" to terminate the session and report the amount of used credit.

Step D-6. The OCS deducts the amount of the used credit from the account. Then the OCS acknowledges the reception of the CCR message by sending a CCA message with type "TERMINATE_REQUEST". This message may contain the total cost information of the service.

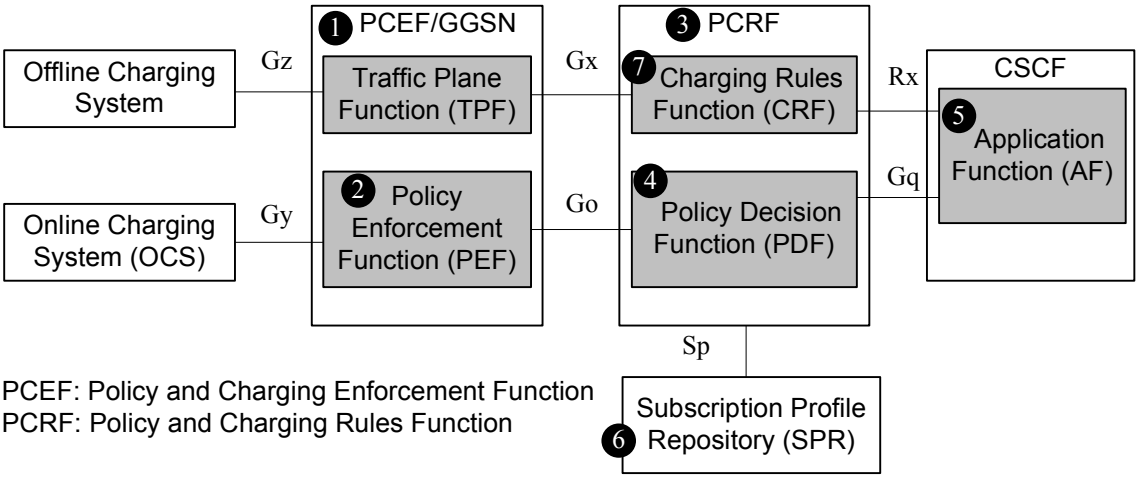


Figure 1.4: PCC architecture for IMS service

1.3 Policy and Charging Control

In this session, we describe the Policy and Charging Control (PCC) architecture of the IMS. In 3GPP R5, the QoS control in IMS/GPRS is realized by the *Session-Based Local Policy* (SBLP) [10]. The SBLP utilizes the *Policy Decision Function* (PDF; see Figure 1.4 (4)) to make policy decisions based on session and media-related information obtained from the CSCF. In other words, the QoS policy is controlled by the PDF, where the policy rules can be formulated based on static information (such as the subscription profile), dynamic information, and the available resources. The combination of such policy rules, once met for a service request, can trigger a desired action (such as allowing the service with the requested bandwidth). This policy rule framework allows the telecom operators to deploy service logic while optimally utilizing the network resource. Specifically, by configuring the policy stored in the PDF, telecom operators are able to implement the QoS policy control flexibly for different applications in various IP networks.

The 3GPP R7 integrates the QoS policy and charging rules to reduce the signaling costs

between network nodes (e.g., GGSN and CSCF) and charging nodes (e.g., *Charging Rules Function* (CRF; Figure 1.4 (7)) and PDF) [21]. The PCC architecture is shown in Figure 1.4. According to the classification of a subscriber, the type of the application to be accessed by the subscriber, and the local control QoS policy defined by the telecom operator, the IMS manages and controls the IP network resources (e.g., the allocated bandwidth) to the application and defines its priority. The *Policy and Charging Rules Function* (PCRF; Figure 1.4 (3)) is responsible for providing IMS charging rules and making policy decisions. In order to make decisions, the PCRF receives information from *Application Function* (AF; Figure 1.4 (5)) and the *Subscription Profile Repository* (SPR; Figure 1.4 (6)). The AF (a standalone application server or being implemented in the CSCF) provides the PCRF with information obtained from the session signaling over the Diameter Rx interface. The SPR provides the PCRF with QoS related information about the user's subscription over the Sp interface.

The PCRF includes the PDF and the CRF. The QoS policy decisions made by the PDF are based on charging-related information (charging rules) and the service information provided from the CRF. In this way, the charging rules are consistent with the QoS policy. The QoS decisions are carried out by the *Policy and Charging Enforcement Function* (PCEF; Figure 1.4 (1)). The PCEF is a logical function implemented in a gateway (e.g., GGSN). This function includes the *Traffic Plane Function* (TPF) and the *Policy Enforcement Function* (PEF; Figure 1.4 (2)). The TPF provides bearer session information (e.g., the user identity, the negotiated QoS and the network-related information) to the CRF. The PEF is responsible for QoS control of the IP service flows. The PCRF communicates with the PCEF over the Diameter-based Gx interface for providing QoS information. Based on the QoS information, the PCEF sets up an aggregate for a service session (i.e., the requirements on the aspects of a connection, such as service response time, bandwidth limit, and so on). The PCEF uses the aggregate to control

downlink/uplink traffic for each bearer service session, i.e., for downlink scheduling and uplink policing.

The offline and online charging systems interact with PCEF for online credit control and the collection of offline charging information through Gz and Gy interfaces, respectively.

1.4 Organization of the Dissertation

Policy and charging are among the most important activities in telecommunications operation today. Based on the above discussion, we investigate the architecture and performance issues of policy and charging in the UMTS/IMS network. This dissertation contains four chapters in addition to this introductory chapter. Details for each chapter are described below.

Chapter 2 and 3 discuss charging issues in network level. In Chapter 2, we study the online credit reservation procedure for prepaid users in the UMTS OCS. If the assigned credit units are consumed before the session is completed, a reserve units operation is executed to obtain more credit units from the OCS. During the RU operation, packet delivery is suspended until extra credit units are granted from the OCS. To avoid session suspension during credit reservation, we propose the credit pre-reservation mechanism that reserves credit earlier before the credit at the GGSN is actually depleted. Analytic and simulation models are developed to investigate the performance of this credit pre-reservation mechanism. Our study provides guidelines to set up the parameters for our proposed mechanism.

In Chapter 3, we design and implement a testbed to investigate the PCC system in *Long Term Evolution* (LTE). Specifically, we use the IMS call service as an example to demonstrate how to implement an advanced mobile service with PCC in our testbed. The PCC system can handle the application level session information (e.g., information obtained from the Application Server)

and the network level bearer information (e.g., information obtained from the core network) dynamically. The corresponding performance measurement (e.g., the signaling delays) can be obtained through our testbed.

In Chapter 4, we focus on charging issues in application level. IMS provides an infrastructure for service-oriented architecture (SOA) development through a service platform based on Parlay X, which is an open web service of telecommunication functionality. Through Parlay X API, application developers can create new services without knowing the details of the telecommunication network (e.g., charging protocol). We design and implement the "group accounting system" service as an example to present how to use standard-based web service APIs to create a charging application in our platform.

Finally, Chapter 5 concludes this dissertation and gives the future directions of this work.

Chapter 2

Credit Pre-reservation Mechanism for UMTS Prepaid Service

The advanced mobile telecommunications operation incorporates data applications (specifically, mobile Internet applications [50, 40]) with real-time control and management, which can be archived by the *Online Charging System* (OCS; Figure 1.1 (14)). Such convergence is essential to mitigate fraud and credit risks, and provide more personalized advice to users about charges and credit limit controls [45, 47, 46]. The OCS allows simultaneous prepaid and postpaid sessions to be charged in real-time [25]. Through online charging, the operator can ensure that credit limits are enforced and resources are authorized on a per-transaction basis.

The OCS assigns some prepaid credit units to the Diameter Credit Control (DCC) client (e.g., GGSN; Figure 1.1 (5)) for a user session. These credit units are decremented at the GGSN in real-time based on either the traffic volume or the duration time. After the assigned credit units are consumed, the GGSN may execute the reserve units (RU) operation to ask for more credit from the OCS.

In the DCC mechanism described in Section 1.2.2, if the credit of the user account at the

OCS is depleted, the prepaid session is forced to terminate. During the RU operation, packet delivery is suspended until extra credit units are granted from the OCS. Delayed processing of user packets at the GGSN may seriously degrade the quality of service. To avoid suspension of packet delivery, this chapter proposes the *credit pre-reservation mechanism* (CPM) that reserves extra credit earlier before the credit units at the GGSN are actually depleted. Analysis and simulation experiments are conducted to investigate the performance of the mechanism. Our study indicates that the CPM can significantly improve the performance of the OCS prepaid mechanism.

2.1 Credit Pre-reservation Mechanism

In the *credit pre-reservation mechanism* (CPM), we define a threshold δ. When the amount c of the remaining credit for the session is not larger than δ (i.e., $c \leq \delta$), the GGSN conducts an RU operation to request extra credit from the OCS. During the RU operation, the GGSN continues to process the user packets. Hopefully, the GGSN will receive the extra amount θ of credit units before $c = 0$, and therefore the user packets need not be buffered (i.e., they are not suspended for processing) at the GGSN. Figure 2.1 illustrates the flowchart of CPM, which modifies Steps D-3 to D-5 in Figure 1.3 as follows:

Step C-1. The session initiates by executing Steps D-1 and D-2 in Figure 1.3.

Step C-3a. The GGSN delivers the user packets and deducts the reserved credit units.

Step C-3b. If the processed packet is the last one of the service session, then Step C-6 is executed to terminate the session (the session is successfully completed). Otherwise, the execution proceeds to Step C-3c.

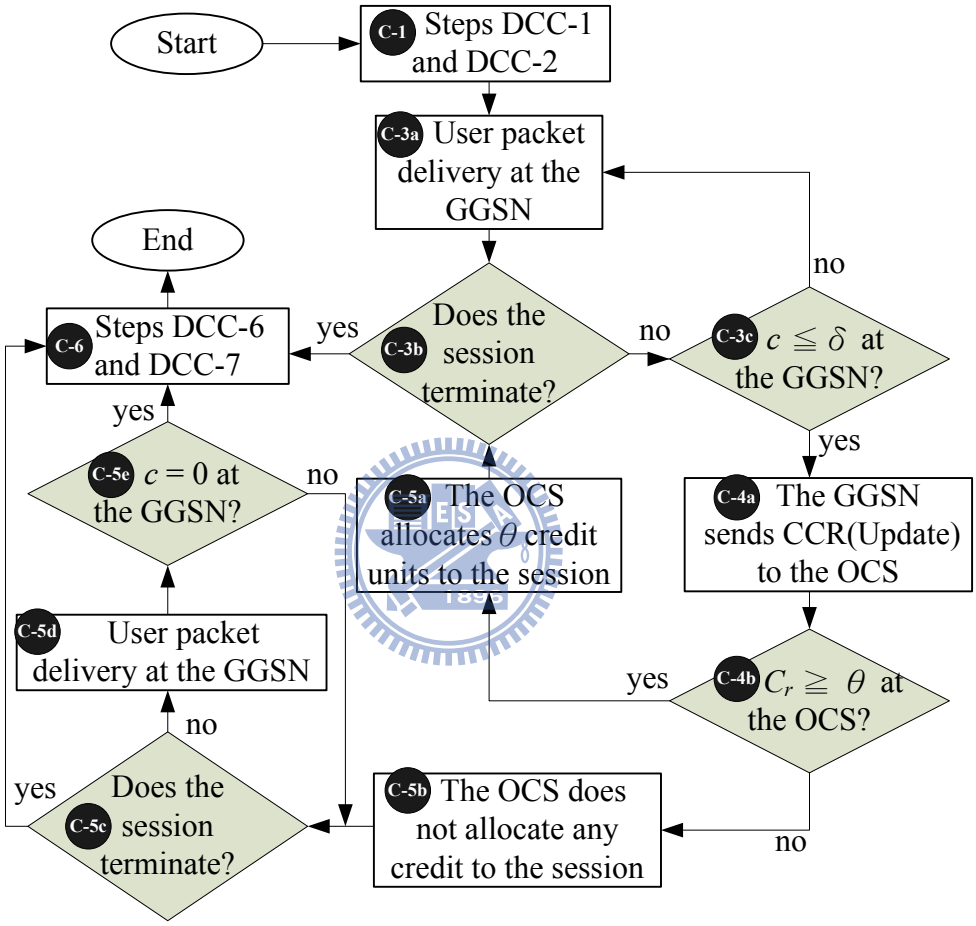


Figure 2.1: Flowchart of the CPM (Steps C-1 and C-6 refer to Steps D-1, D-2, D-6 and D-7 in Figure 1.3

Step C-3c. Let δ be the CPM threshold. If $c \leq \delta$, Step C-4a is executed. Otherwise, the execution proceeds to Step C-3a.

Step C-4a. The GGSN sends a `CCR` message with type UPDATE_REQUEST to request for additional credit. During the RU operation, if $c > 0$, the user packets are continuously delivered at the GGSN. When $c = 0$, the session is suspended and the newly arriving packets are buffered.

Step C-4b. If the OCS does not have enough credit units (i.e., $C_r < \theta$), Step C-5b is executed. Otherwise, Step C-5a is executed.

Step C-5a. The OCS sends the `CCA` message to the GGSN to indicate that extra amount θ of credit units have been reserved for the session. Then the execution proceeds to Step C-3b. If the last packet arrives during the RU operation, the termination operation (Steps C-3b and C-6) is executed after the GGSN has received the `CCA` message. (This is called *delayed termination*.) In this case, the session is successfully completed.

Step C-5b. The OCS sends the `CCA` message to the GGSN. This message indicates that no credit is reserved for the session.

Step C-5c. If the previously processed packet is the last one of the session, then the session is successfully completed. Step C-6 is executed.

Step C-5d. The GGSN continues to deliver the user packets.

Step C-5e. If $c = 0$, then the session is forced to terminate, and Step C-6 is executed. Otherwise, the execution proceeds to Step C-5c.

Step C-6. The session terminates by executing Steps C-6 and C-7 in Figure 1.3.

In the CPM, if δ is set too small, then the credit units for a session are likely to be depleted and the session must be suspended during the RU operation. On the other hand, if δ is set too large, many credit units are reserved in the active sessions, and the credit in the OCS is consumed fast. In this case, an incoming session has less chance to be served, and an in-progress session is likely to be force-terminated. Therefore, it is important to select an appropriate δ value to "optimize" the CPM performance.

2.2 Analytic Model for the CPM

In this section, we describe an analytic model to investigate the CPM performance. We assume that the prepaid session arrivals for a user form a Poisson process with rate γ. The inter-arrival time between two packet arrivals has the mean $1/\lambda$. The round-trip transmission delay for the RU operation (i.e., the round-trip message delay for the `CCR` and `CCA` message pair) has the mean $1/\mu$. An arrival packet is the last one of the session with probability α; in other words, the session continues with probability $1-\alpha$, and the expected number of packets delivered in a session is $1/\alpha$.

Initially, a user has C credit units at the prepaid account in the OCS. Without loss of generality, we assume that each user packet consumes one credit unit. Define a *low credit* (LC) *period* as an interval such that during this interval, $c \leq \delta$ for a session. At the beginning of an LC period, the session initiates an RU operation. If more than θ packets arrive during this RU operation, then $\theta - \delta$ packets will be buffered at the GGSN. Consequently, at the end of the RU operation, another RU operation must be issued to obtain more credit units to absorb the buffered packets and to ensure that $c > \delta$ after the buffer is empty. Before an LC period ends, the RU operation may be executed for several times until the session has reserved more than δ

credit units. The output measures investigated in our study are listed below.

B: the average number of packets buffered during an RU operation

W: the average packet waiting time

P_r: the probability that during an LC period, two or more RU operations are executed

P_{nc}: the probability that a session is not completely served; i.e., the probability that a new session request is blocked or an in-progress session is forced to terminate

X_s: the average number of the RU operations performed in a session

To derive P_r and B, we first consider the case where $\alpha = 0$; i.e., a session is never terminated. Let K be the number of packets arriving in one RU operation (excluding the first packet arrival that triggers the RU operation). It is clear that

$$P_r = \Pr[K \geq \theta] \tag{2.1}$$

We assume that an RU operation delay has the Erlang density function $f(t)$ with the shape parameter $b = 2$ and the scale parameter $h = 1/\mu$. (I.e., t is the summation of two Exponential delays. This assumption will be relaxed, and more general distributions will be considered in the simulation model) Therefore the Laplace-Stieltjes Transform $f^*(s)$ of the RU operation delay is

$$f^*(s) = \left(\frac{\mu}{\mu + s}\right)^2 \tag{2.2}$$

For $\alpha = 0$, the probability that $K = k$ can be calculated as follows:

$$\Pr[K = k, \alpha = 0] = \int_{t=0}^{\infty} \left[\frac{(\lambda t)^k}{k!} \right] e^{-\lambda t} f(t) dt \tag{2.3}$$

$$= \left(\frac{\lambda^k}{k!} \right) \int_{t=0}^{\infty} t^k e^{-\lambda t} f(t) dt \tag{2.4}$$

$$= \left(\frac{\lambda^k}{k!} \right) (-1)^k \left[\frac{d^k f^*(s)}{ds^k} \right] \Bigg|_{s=\lambda} \tag{2.5}$$

$$= \left(\frac{\lambda^k}{k!} \right) (-1)^k \left[\frac{d^k}{ds^k} \left(\frac{\mu}{\mu + s} \right)^2 \right] \Bigg|_{s=\lambda} \tag{2.6}$$

$$= \frac{\lambda^k (k+1) \mu^2}{(\lambda + \mu)^{k+2}} \tag{2.7}$$

In (2.3), the RU operation delay is t with the probability $f(t)dt$. During period t, there are k packet arrivals following the Poisson distribution with the rate λ. Equation (2.5) is derived from (2.4) using Rule P.1.1.9 in [49]. Substitute (2.2) in (2.5), we obtain (2.6).

Now we consider the case when $\alpha > 0$. If a session is terminated during an RU operation, $\Pr[K = k]$ is derived by considering the following cases:

(I) When $k = 0$, we have $\Pr[K = 0] = \Pr[K = 0, \alpha = 0]$

(II) When $k > 0$, there are two subcases:

(IIa) There are exactly k packet arrivals during an RU operation (with probability $\Pr[K = k, \alpha = 0]$) and the session is not terminated by any of the first $k-1$ packets (with the probability $(1 - \alpha)^{k-1}$). Note that the k-th packet can be the last one of the session.

(IIb) There are more than k packet arrivals during an RU operation if the session is never terminated (with probability $\sum_{i=k+1}^{\infty} \Pr[K = i, \alpha = 0]$), and the session is actually terminated at the k-th packet arrival (with probability $(1 - \alpha)^{k-1} \alpha$).

Based on the above cases, $\Pr[K=k]$ is derived as

$$\Pr[K=k] = \begin{cases} \Pr[K=0,\alpha=0] & ,k=0 \quad (2.8) \\ \Pr[K=k,\alpha=0]\,(1-\alpha)^{k-1} & \\ \quad + \sum_{i=k+1}^{\infty} \Pr[K=i,\alpha=0]\,(1-\alpha)^{k-1}\,\alpha & ,k>0 \quad (2.9) \end{cases}$$

From (2.7), (2.8) can be derived as

$$\Pr[K=0] = \left(\frac{\mu}{\lambda+\mu}\right)^2 \tag{2.10}$$

For $k>0$, Equation (2.9) is simplified as

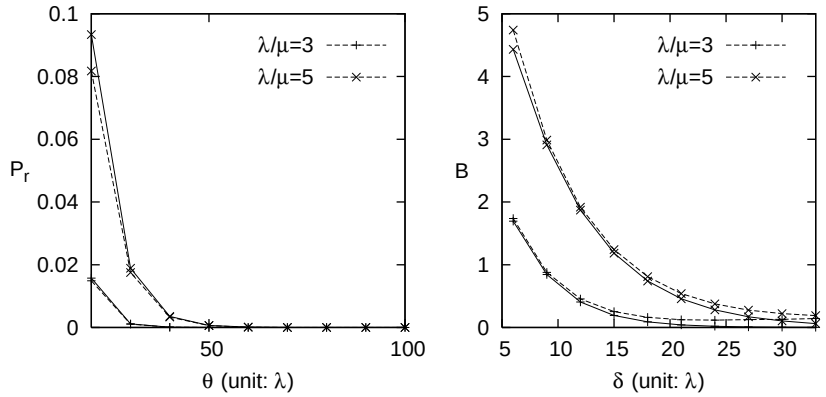
$$\begin{aligned} \Pr[K=k] &= \left[\frac{\lambda^k(k+1)\mu^2}{(\lambda+\mu)^{k+2}}\right](1-\alpha)^{k-1} \\ &\quad + \sum_{i=k+1}^{\infty}\left[\frac{\lambda^i(i+1)\mu^2}{(\lambda+\mu)^{i+2}}\right](1-\alpha)^{k-1}\,\alpha \\ &= \left[\frac{(1-\alpha)^{k-1}\lambda^k}{(\lambda+\mu)^{k+2}}\right] \\ &\quad \left\{(k+1)\mu^2 + \alpha\lambda\left[(k+2)\mu+\lambda\right]\right\} \end{aligned} \tag{2.11}$$

When $\theta>0$, from (2.1) and (2.11), P_r is derived as

$$\begin{aligned} P_r &= \Pr[K\geq\theta] \\ &= \sum_{k=\theta}^{\infty}\left[\frac{(1-\alpha)^{k-1}\lambda^k}{(\lambda+\mu)^{k+2}}\right]\left\{(k+1)\mu^2+\alpha\lambda\left[(k+2)\mu+\lambda\right]\right\} \\ &= \left[\frac{\lambda^\theta(1-\alpha)^{\theta-1}}{(\lambda+\mu)^{\theta+1}}\right][\mu(\theta+1)+\lambda] \end{aligned} \tag{2.12}$$

When $\theta=0$,

$$P_r = \Pr[K\geq 0] = \left(\frac{\mu}{\mu+\lambda}\right)^2 + \sum_{k=1}^{\infty}\Pr[K\geq k] = 1 \tag{2.13}$$



(a) Effect on P_r (b) Effect on B

Figure 2.2: Validation of simulation and analytic results on P_r and B ($\alpha = 0.01$, $\gamma/\mu = 1/20$, and $\delta = 0.3\theta$; Solid curves: analytic results; dashed curves: simulation results)

The expected number B of buffered packets is derived as follows: When the number k (of packet arrivals during an RU operation) is no less than the threshold δ (i.e., $k \geq \delta$), the session will have $k - \delta$ buffered packets. Therefore, from (2.11)

$$\begin{aligned} B &= \sum_{k=\delta}^{\infty} (k-\delta) \Pr[K=k] \\ &= \sum_{k=\delta}^{\infty} (k-\delta) \left[\frac{(1-\alpha)^{k-1} \lambda^k}{(\lambda+\mu)^{k+2}} \right] \\ &\quad \left\{ (k+1)\mu^2 + \alpha\lambda \left[(k+2)\mu + \lambda \right] \right\} \\ &= \left[\frac{(1-\alpha)\lambda}{\lambda+\mu} \right]^{\delta+1} \\ &\quad \left[\frac{\delta\mu^2 + \alpha\delta\lambda\mu + 2\mu^2 + 2\lambda\mu + \alpha\lambda\mu + \alpha\lambda^2}{(1-\alpha)(\mu+\alpha\lambda)^2} \right] \end{aligned} \tag{2.14}$$

The purpose of the analytic model is twofold: First, it partially verifies that the simulation model is correct. Second, it sheds light on the effects of the input parameters on P_r and B. Our simulation model is based on an event-driven approach widely adopted in mobile network studies, and the details are elaborated in Appendix A.

Equations (2.12) and (2.14) validate against the simulation experiments as illustrated in

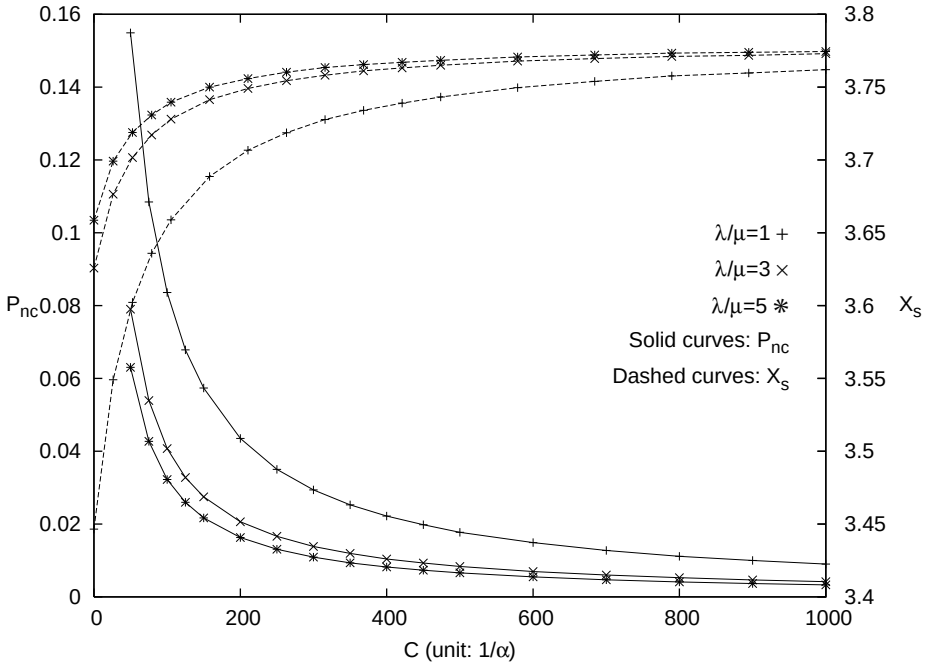


Figure 2.3: Effects of λ/μ and C on P_{nc} and X_s (α=0.01,γ/μ=1/20, δ=0.3θ, and θ=50λ)

Figure 2.2. In this figure, the dashed curves represent the simulation results, and the solid curves represent the analytic results. These curves indicate that the analytic and the simulation results are consistent.

2.3 Numerical Examples

This section uses numerical examples to investigate the performance of the CPM. For the presentation purpose, we assume that the packet termination probability is $\alpha = 0.01$ and the session arrival rate (normalized by the message delivery rate) is $\gamma = \mu/20$. For other α and γ/μ values, we observed similar results, which are not presented in this chapter. In Figures 2.2 and 2.3, the packet arrivals in a session have a Poisson distribution and the RU operation delay has an Erlang distribution. These Expontial-like assumptions are relaxed in Figures 2.4 and 2.5 by

considering the Pareto and the Gamma distributions. The effects of the input parameters λ/μ, C, θ and δ are described as follows.

Effects on P_r. Figure 2.2 (a) shows how P_r is affected by θ and λ/μ. From (2.12) and (2.13), we have

$$\lim_{\theta \to 0} P_r = 1 \quad \text{and} \quad \lim_{\theta \to \infty} P_r = 0 \tag{2.15}$$

Therefore, it is obvious that P_r is a decreasing function of θ.

When λ/μ is very small or vary large, we have

$$\lim_{\lambda/\mu \to 0} P_r = (1-\alpha)^{\theta-1} \quad \text{and} \quad \lim_{\lambda/\mu \to \infty} P_r = 0 \tag{2.16}$$

When λ/μ increases, more packets arrive during an RU operation. When more than θ+δ packets arrive during one RU operation, an extra RU operation will be immediately executed. Consequently, P_r increases. Therefore P_r is an increasing function of λ/μ.

Effects of λ/μ. Figure 2.2 (b) shows that B is increasing functions of λ/μ. From (2.14), we have

$$\lim_{\lambda/\mu \to 0} B = 0 \quad \text{and} \quad \lim_{\lambda/\mu \to \infty} B = \frac{(1-\alpha)^{\delta}}{\alpha} \tag{2.17}$$

When λ/μ increases, it is likely that more than δ packets will arrive during the interval of the RU operation. Note that W correlates positively with B. Therefore both B and W increase as λ/μ increases.

When $\lambda/\mu \to \infty$, we found that the B value in (2.17) is higher than the simulation result (not shown in this Figure), which is explained as follows: When $\lambda/\mu \to \infty$, all packets for a session will arrive before the end of the first RU operation, and therefore the B value in (2.17) is determined by α. In simulation, the session is always in the "low-credit" status

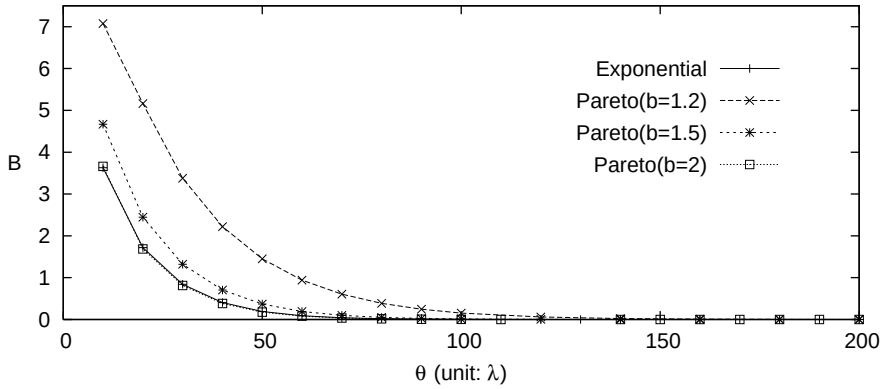
(in the LC period), and every time an RU operation is performed, the number of buffered packets at the end of the operation is reduced. Therefore the expected value B is smaller than that shown in (2.17). To avoid buffer overflow, the B value in (2.17) should be considered in the system setup.

Figure 2.3 shows that P_{nc} increases as λ/μ decreases. Since α is fixed, when λ/μ decreases, the session holding times become longer, and it is likely that more new sessions will arrive during the holding time of an existing session. Therefore, when λ/μ decreases, more sessions will exist at the same time. Suppose that the credit in the OCS suffices to support these sessions if they are sequentially delivered. It is clearly that the OCS may not be able to support these sessions if they are delivered simultaneously. In this case, a newly incoming session is rejected because the credit in the OCS is depleted (while there are unused credit units held in the multiple in-progress sessions). Therefore, P_{nc} increases as λ/μ decreases.

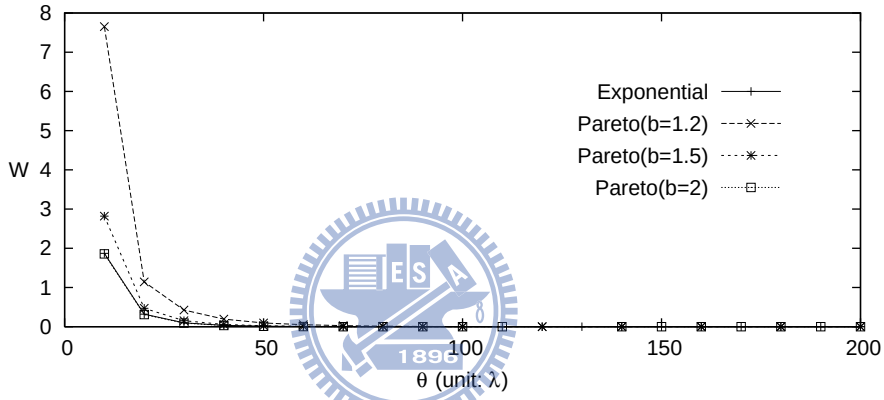
In Figure 2.3, X_s is a decreasing function of P_{nc} because the number of RU operations performed in a force-terminated session is less than that in a complete session. Therefore, X_s increases as λ/μ increases.

Effects of *C*. Figure 2.3 shows that the output measures (P_{nc} and X_s) are only affected by the "end effect" of C. As C increases, it is more likely that the remaining credit units in the OCS suffice to support one RU operation and such end effect becomes insignificant. Similar to the λ/μ impact, P_{nc} is a decreasing function of *C*, and X_s is an increasing function of *C*. Figure 2.3 indicates that when *C* is sufficiently large (e.g. $C \geq 600/\alpha$), the end effect of C can be ignored. Same phenomenon is observed for B and W, and the results are not shown.

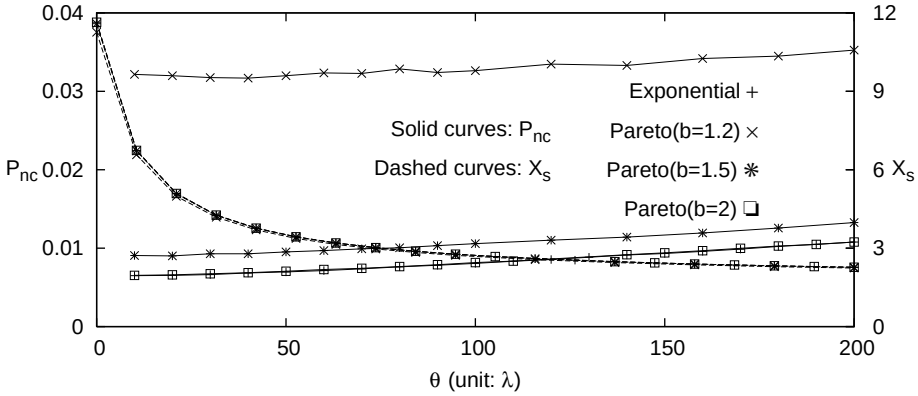
Effects of θ. Figure 2.4 intuitively shows that B, W, and X_s are decreasing functions of θ.



(a) Effect on B



(b) Effect on W



(c) Effect on P_{nc} and X_s

Figure 2.4: Effects of θ and the packet interarrival time distribution (α=0.01, γ/μ=1/20, δ=0.3θ, $C = 600/\alpha$, and λ/μ=3)

Since $\delta = 0.3\theta$ in Figure 2.4 (b), from (2.13) and (2.14), we have

$$\lim_{\theta\to\infty} B = \lim_{\delta\to\infty} B = 0 \quad \text{and}$$
$$\lim_{\theta\to 0} B = \lim_{\delta\to 0} B = \frac{\lambda\left(2\mu+\alpha\lambda\right)}{\left(1-\alpha\right)\left(\mu+\alpha\lambda\right)^2} \tag{2.18}$$

The non-trivial result is that there is a threshold θ value ($\theta \approx 100\lambda$ in Figure 2.4) such that beyond this threshold value, increasing θ does not improve the performance. On the other hand, Figure 2.4 (c) shows that P_{nc} linearly increases as θ increases. When θ increases, more credit units are reserved in an RU operation, and the credit in the OCS is consumed fast. Therefore, a newly incoming session has less chance to be served, and an in-progress session is likely to be force-terminated.

Effects of packet interarrival time distribution. Figure 2.4 considers the packet arrival times with the Exponential and the Pareto distributions with mean $1/\lambda$. In the Pareto distribution, the shape parameter b describes the "heaviness" of the tail of the distribution. It has been shown that the Pareto distribution with $1 \leq b \leq 2$ can approximate the packet traffic very well [26, 51].

Figure 2.4 shows that B, W and P_{nc} are decreasing functions of b. When b decreases, the tail of the distribution becomes longer, and more long packet interarrival times are observed. Since the mean value $1/\lambda$ is fixed for the Pareto distribution in Figure 2.4, more long packet interarrival times also imply more short packet interarrival times. The number of short interarrival times must be larger than that of long interarrival times because the minimum of the interarrival time is fixed but the maximum of the interarrival time is infinite. Thus, it is likely that more packets will arrive during an RU operation, and B and W increase. With a small b, it is likely that the last session for a user accommodated by the OCS is a very long session. Before the session is completed, new sessions continue

to arrive, and are rejected by the OCS, which contributes to P_{nc}. Therefore P_{nc} increases as b decreases.

Effects of δ. Similar to the effect of θ, Figure 2.5 shows that B and W are decreasing functions of δ (see (2.18)), and P_{nc} is a linearly increasing function of δ. A non-trivial observation is that when $\delta \geq 0.6\theta$, B and W approach to zero. It implies that selecting δ value larger than 0.6θ will not improve the performance. Figure 2.5 (c) shows that X_s is an increasing function of δ. When the amount of the credit in a session is less than δ, a `CCR` message is sent to the OCS. Therefore, for a fixed θ, when δ is increased, X_s increases.

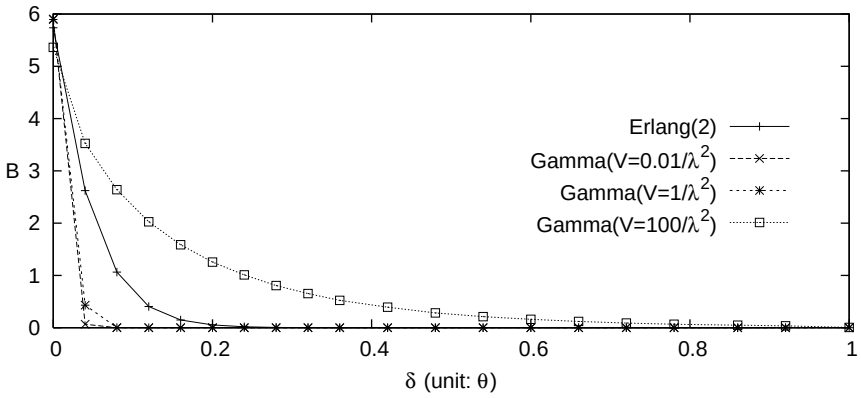
Effects of RU operation delay distribution. Figure 2.5 considers the Erlang with $b = 2$ (which is a Gamma distribution with variance $V = 18/\lambda^2$) and Gamma distributed RU operation delays with variances $V = 0.01/\lambda^2, 1/\lambda^2$, and $100/\lambda^2$, respectively.

The figure indicates that P_{nc} and X_s are not significantly affected by the RU operation delay distribution. On the other hand, B and W increase as V increases. As V increases, more long and short RU operation delays are observed. In long RU operation delays, it is likely that more than δ packets arrive. Therefore the packets are more likely to be buffered and delayed processed.

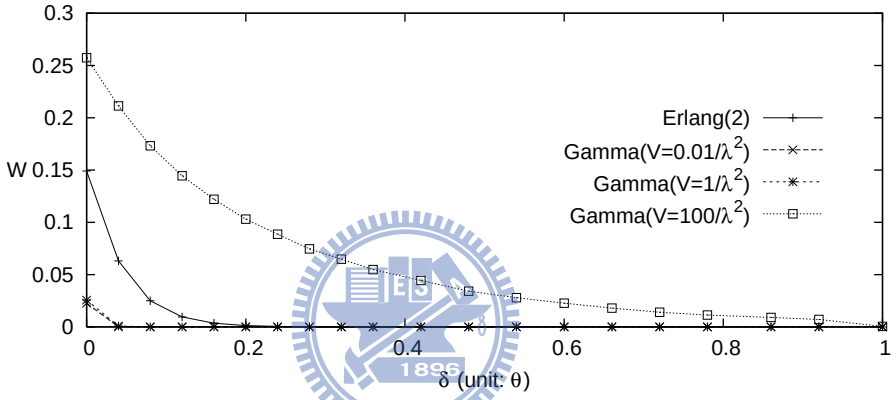
2.4 Conclusion

In this chapter, we investigated the prepaid services for the UMTS network where multiple prepaid and postpaid sessions are simultaneously supported for a user. We described the prepaid network architecture based on UMTS, and proposed the credit pre-reservation mechanism (CPM) that reserves extra credit earlier before the credit at the GGSN is actually depleted.

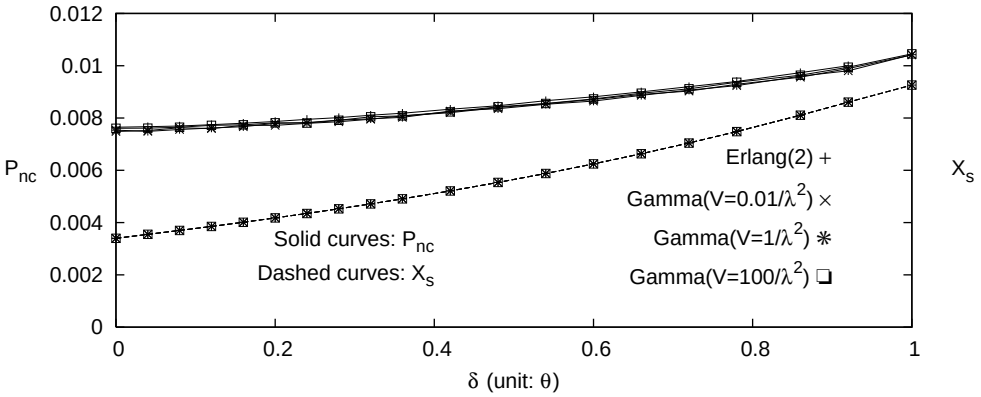
An analytic model was developed to compute the average number B of packets buffered



(a) Effect on B



(b) Effect on W



(c) Effect on P_{nc} and X_s

Figure 2.5: Effects of δ and the RU operation delay distribution (α=0.01, γ/μ=1/20, θ=100λ, $C = 600/\alpha$, λ/μ=3, and the packet arrival times have the Pareto distribution with the mean $1/\lambda$ and $b = 2$)

during a reserve units (RU) operation and the probability P_r that more than one RU operation is executed during a low credit (LC) period. Simulation experiments are conducted to investigate the performance of CPM. We have the following observations:

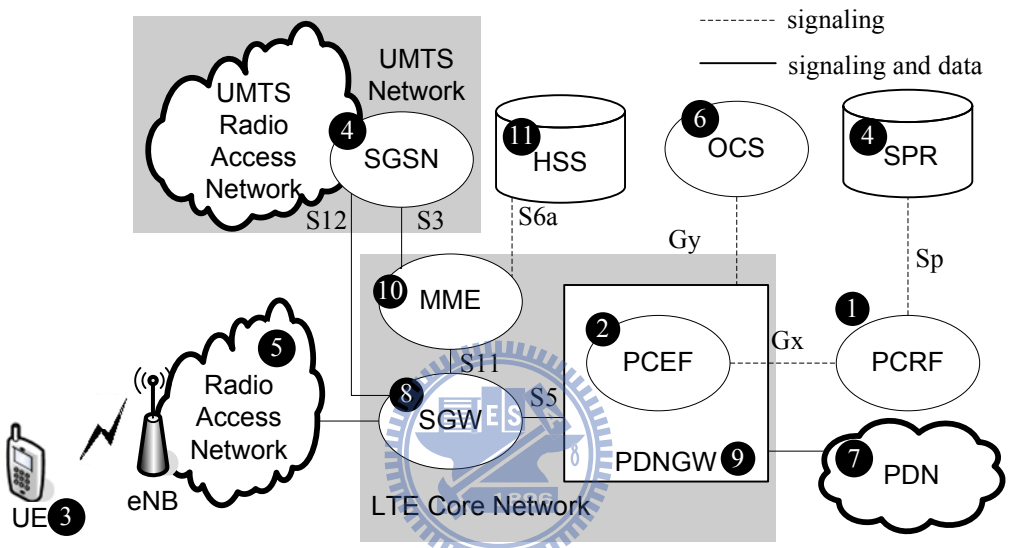
- B and W increase as λ/μ increases. The probability P_{nc} that a session is not completely served decreases as λ/μ increases. The average number X_s of RU operations performed in a session is an increasing function of λ/μ.
- B, W, X_s and P_{nc} are only affected by the end effect of C. When C is sufficiently large (e.g., $C \geq 600/\alpha$, where α is the probability that an arrival packet is the last one of the session), the end effect can be ignored.
- B, W and X_s decrease but P_{nc} increases as θ increases. There is a threshold θ value (e.g., $\theta \approx 100\lambda$) such that beyond this threshold value, increasing θ does not improve the CPM performance.
- B and W decrease but X_s increases as δ increases. When δ is large (e.g., $\delta \geq 0.6\theta$), both B and W approach zero.
- P_r increases when λ/μ increases or θ decreases.
- B, W and P_{nc} increase as the tail of the packet arrival time distribution becomes longer.
- B and W increase as the variance of the RU operation delay increases.

Our study provides guidelines to select the CPM parameters. Specifically, it is appropriate to select $C \geq 600/\alpha$, $\theta \approx 100\lambda$, and $\delta \approx 0.6\theta$.

Chapter 3

Policy and Charging Control System for Advanced Mobile Services

The 3rd Generation Partnership Project (3GPP) Release 8 introduces *Long Term Evolution* (LTE) that is a set of enhancements to the *Universal Mobile Telecommunications System* (UMTS). Since LTE offers higher throughput and lower latency, *Policy and Charging Control* (PCC) has become a major issue for guaranteed security, *Quality of Service* (QoS) and flexible charging. We design and implement a testbed to investigate the impact of PCC on advanced mobile service deployment. This experimental testbed provides web-based user interface for an administrator (e.g., the telecom operator or a subscriber) to conveniently set up the service admission policy (e.g., for online charging budget). These policy settings are converted to PCC rules and are used to ensure the QoS.



PDN: Packet Data Network
PDNGW: PDN Gateway
SPR: Subscription Profile Repository
UE: User Equipment
PCRF: Policy and Charging Rules Function
PCEF: Policy and Charging Enforcement Function
MME: Mobility Management Entity
SGW: Serving Gateway
HSS: Home Subscriber Server
SGSN: Serving GPRS Support Node
eNB: e Node B
OCS: Online Charging Server

Figure 3.1: Policy and Charging Control Architecture in LTE Network

3.1 Policy and Charging Control Architecture in LTE Network

As illustrated in Figure 3.1, the LTE core network (also known as *Evolved Packet Core* (EPC)) includes the *Mobility Management Entity* (MME; Figure 3.1 (10)), the *Serving Gateway* (SGW; Figure 3.1 (8)) and the *Packet Data Network Gateway* (PDNGW; Figure 3.1 (2)). The MME manages service sessions authentication (by interacting with the *Home Subscriber Server*; HSS, Figure 3.1 (11) or Figure 1.1 (6)), paging, roaming and bearer management. The MME also provides mobility management between LTE and UMTS networks (i.e., it connects to the *Serving GPRS Support Node* (SGSN); Figure 3.1 (4) and Figure 1.1 (4)) with the S3 interface. The SGW plays a role similar to the SGSN in UMTS, which routes and forwards user data packets between the gateway node (PDNGW or *Gateway GPRS Support Node* (GGSN)) and the radio access network. The PDNGW provides connectivity from the *User Equipment* (UE; Figure 3.1 (3)) to packet data network (PDN; Figure 3.1 (7) or Figure 1.1 (7)). The PDNGW is similar to GGSN (Figure 1.1 (5)) in UMTS.

When a UE initiates a service session, the session is connected to the PDNGW through the radio access network (Figure 3.1 (5)) and the SGW [19]. Before establishing the connection, the PDNGW requests the PCC rules of the UE from the *Policy and Charging Rules Function* (PCRF; Figure 3.1 (1)). The PCRF specifies the PCC rules for a service session so that policy enforcement and charging management can be performed in the LTE network. The *Policy and Charging Enforcement Function* (PCEF; Figure 3.1 (9)) is implemented at the PDNGW. According to the subscriber's charging plan, the network service to be accessed by the UE and the control policy defined by the telecom operator, the PCRF makes policy decisions and provides the PCC rules to the PCEF through the Gx interface (see Chapter 9 in [36]). The

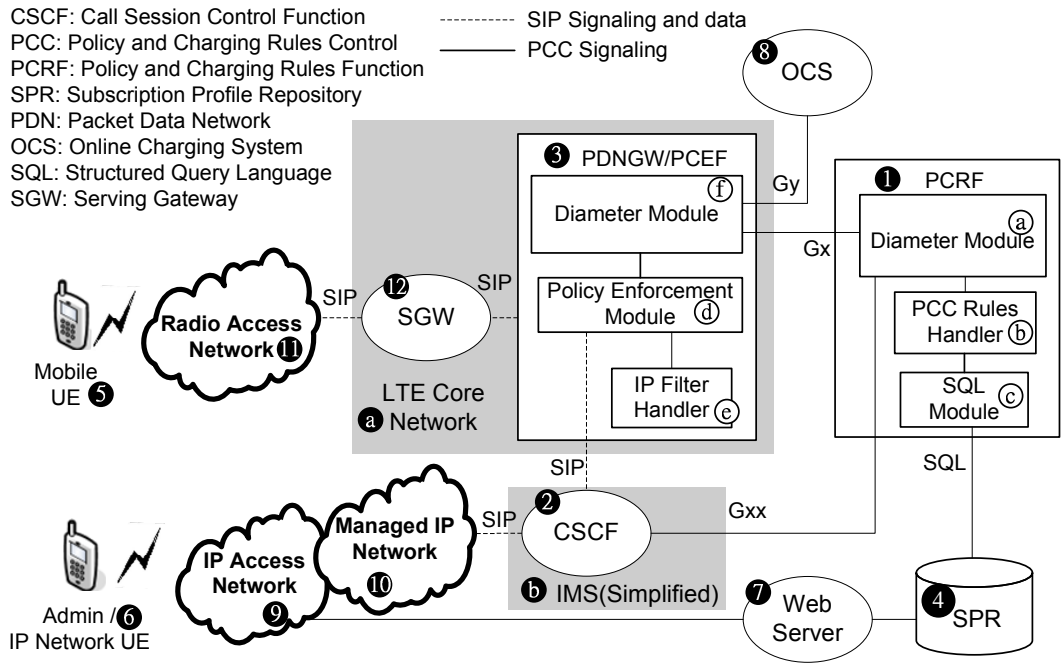


Figure 3.2: The Block Diagram for the PCC Testbed Implementation

Gx interface as well as Gg and Gxx [5, 7] are implemented by the Diameter protocol [24, 30] for *Authentication, Authorization and Accounting* (AAA). This testbed allows operators to experiment on advanced mobile services with PCC control without implementing these services in the commercial telecom system. This chapter is organized as follows: we first present the PCC testbed and its detailed design, and we use an IMS call control service as an example to show how our PCC testbed works. Finally, we provide measurement results to investigate the performance of the PCC testbed.

3.2 The PCC Testbed

This section describes a PCC testbed for advanced mobile services. We first describe the functional blocks of PCC, and use the IMS call control service as an example to illustrate the PCC

capabilities. In Figure 3.2, the dashed lines represent PCC signaling (e.g., Diameter and SQL messages) and the solid lines represent the SIP signaling and data. The UEs can reside in the IP access network (i.e., Figure 3.2 (9)) or the mobile network (e.g., *Radio Access Network*; Figure 3.2 (11)). The IP network UE connects to the IMS network (Figure 3.2 (c)) through managed IP network (QoS guaranteed private Internet; Figure 3.2 (10)). In our testbed, the administrator (e.g., the telecom operator or a subscriber) configures the PCC settings through a web-based portal (i.e., the *Web Server*; Figure 3.2 (7)) offered by the operator. The PCC settings are converted to PCC rules for access control in our testbed. The access control is handled based on the call duration time limit (e.g., a subscriber is only allowed to use the cell phone for 30 minutes per day) and the bandwidth limit. Through appropriate PCC settings, the PCC services (e.g., the IMS call service) can be easily deployed.

3.2.1 The Functional Blocks of the PCC Testbed

The functional modules of PCRF (Figure 3.2 (1) or Figure 1.4 (1)) and PCEF (Figure 3.2 (3) or Figure 1.4 (2)) in our testbed can be found in Figure 3.2. The PCRF includes the Diameter Module (Figure 3.2 (a)), the PCC Rules Handler (Figure 3.2 (b)) and the SQL Module (Figure 3.2 (c)). The PCRF Diameter Module implements the Gx and Rx interfaces [8]. This module communicates with the IMS *Call Session Control Function* (CSCF) (Figure 3.2 (2)) for delivering the service request over the Diameter Rx `Authorize Authenticate Request/Answer (AAR/AAA)` messages. The Diameter Module also communicates with the PDNGW (Figure 3.2 (3) or Figure 1.4 (9)) to provide the PCC rules through the Diameter Gx `Credit Control Request/Answer (CCR/CCA)` messages. This module can be reused in the PCEF for the Diameter messages exchange. A Diameter message includes information of the service session, such as the caller ID, serving time, and the callee ID. The PCC

Rules Handler collects the service requests from the CSCF and retrieves the subscriber information in the service request. Then the PCRF uses the subscriber information (e.g., the caller ID) to query the SPR (Figure 3.2 (4)) for retrieving the PCC information of the caller and callee through the SQL Module. The SQL Module queries the SPR to retrieve the subscriber profile by SQL commands. Based on the query results and the subscriber information, the PCC Rules Handler makes the PCC rules for the service session, and sends the rules to PCEF/PDNGW for PCC rule enforcement.

The PCEF is located in PDNGW, which consists of three PCC modules: the Policy Enforcement Module (Figure 3.2 (d)), the Diameter Module (Figure 3.2 (f)) and the IP Filter Handler (Figure 3.2 (e)). The Policy Enforcement Module executes the PCC rules and controls the packet transmission by invoking the IP Filter Handler, e.g., the service session will be forced to terminate if the call duration time limit is reached. When the Policy Enforcement Module decides to terminate a service session, it will send a Session Initiation Protocol (SIP) `BYE` message [42] to the UE. The PCEF Diameter Module implements the Gx and Gy interfaces. The Diameter Module retrieves the PCC rules from the PCRF through Gx and communicates with the Online Charging Server (OCS; Figure 3.2 (8) or Figure 1.4 (6)) through Gy. The IP Filter Handler enforces the rules by the iptables [48], which is included in the Linux kernel. The iptables monitors the incoming and outgoing network packets. By default, the IP Filter Handler blocks every packet. When it receives the PCC rules from the PCRF, the iptables setting will be modified to allow the packet transmission for controlling the session duration or the bandwidth. When the session is terminated, the bandwidth setting will be removed and the iptables restores to its previous setting. In our implementation, we use the following two iptables settings to filter packets.

Call duration control: Initially, we block every Real-Time Protocol (RTP) [43] data packets

```
(1)  public void CallDurationControlRule() {
(2)      Runtime.getRuntime().exec("iptables -N FORWARD -s " +
             subscriberIP + " -j ACCEPT");
(3)      Timer timer = new Timer();
(4)      timer.schedule(new Task(), duration*1000);
         ...
     }
     class Task extends TimerTask {
         public void run(){
(5)          Runtime.getRuntime().exec("iptables -N FORWARD -s " +
                 subscriberIP + " -j DENY");
(6)          // terminate the call service session
         }
     }
```

Figure 3.3: Call Duration Control Program Segment

by default. When the PDNGW receives the PCC rules, the iptables setting function CallDurationControlRule shown in Figure 3.3 will be executed. Line 2 allows all the packets from the subscriber. In the meanwhile, we set a timer to control the call duration (Lines 3 and 4). If the call duration time limit is reached, the previous iptables setting will be removed (Line 5) and the service session will be forced to terminate (Line 6).

```
(1)  public void BandwidthControlRule() {
(2)      Runtime.getRuntime().exec("iptables -N chain" + chainNo);
(3)      Runtime.getRuntime().exec("iptables -A FORWARD -s " +
             subscriberIP + " -j chain" + chainNo);
(4)      Runtime.getRuntime().exec("iptables -A chain" + chainNo + " -s " +
             subscriberIP + "-m limit –limit 10/s –limit-burst 5 -j ACCEPT");
     }
```

Figure 3.4: Bandwidth Control Program Segment

Bandwidth control: Figure 3.4 shows the program segment of the iptables bandwidth control setting. We use the iptables *limit* module (which is included in the linux core) for the bandwidth control. When the PDNGW receives the PCC rules, the iptables setting function BandwidthControlRule shown this figure is executed. Line 2 and 3 create a custom chain (i.e., packet filter) for the subscriber's IP. Line 4 limits the bandwidth of the custom chain to 10 RTP packets per second and the network burst to 5 packets. When the service session is terminated, the iptables setting will be removed and be restored to the previous setting.

The CSCF is responsible for the IMS signaling control and the service information provision. In our implementation, we use openSIPS [39], which is an open source implementation of a SIP server, to develop the CSCF.

The SPR (Figure 3.2 (4)) is a database that stores the subscription profiles (i.e., PCC rules for the subscriber) and other related charging information. The subscription profile can be dynamically configured by the Web Server.

Note that the Diameter Modules in PCRF and PDNGW are developed on the basis of Open Diameter project [38], which is a session based API for RFC 3588. We extended this project to accommodate the Diameter messages described in other specifications (i.e., RFC 4006). More specifically, we developed the C++ classes AAA_CreditControlClient and AAA_CreditControlServer, which are respectively inherited from the AAA_ClientAuthSession and AAA_ServerAuthSession classes in the Open Diameter library [38]. More details can be found in Appendix B.

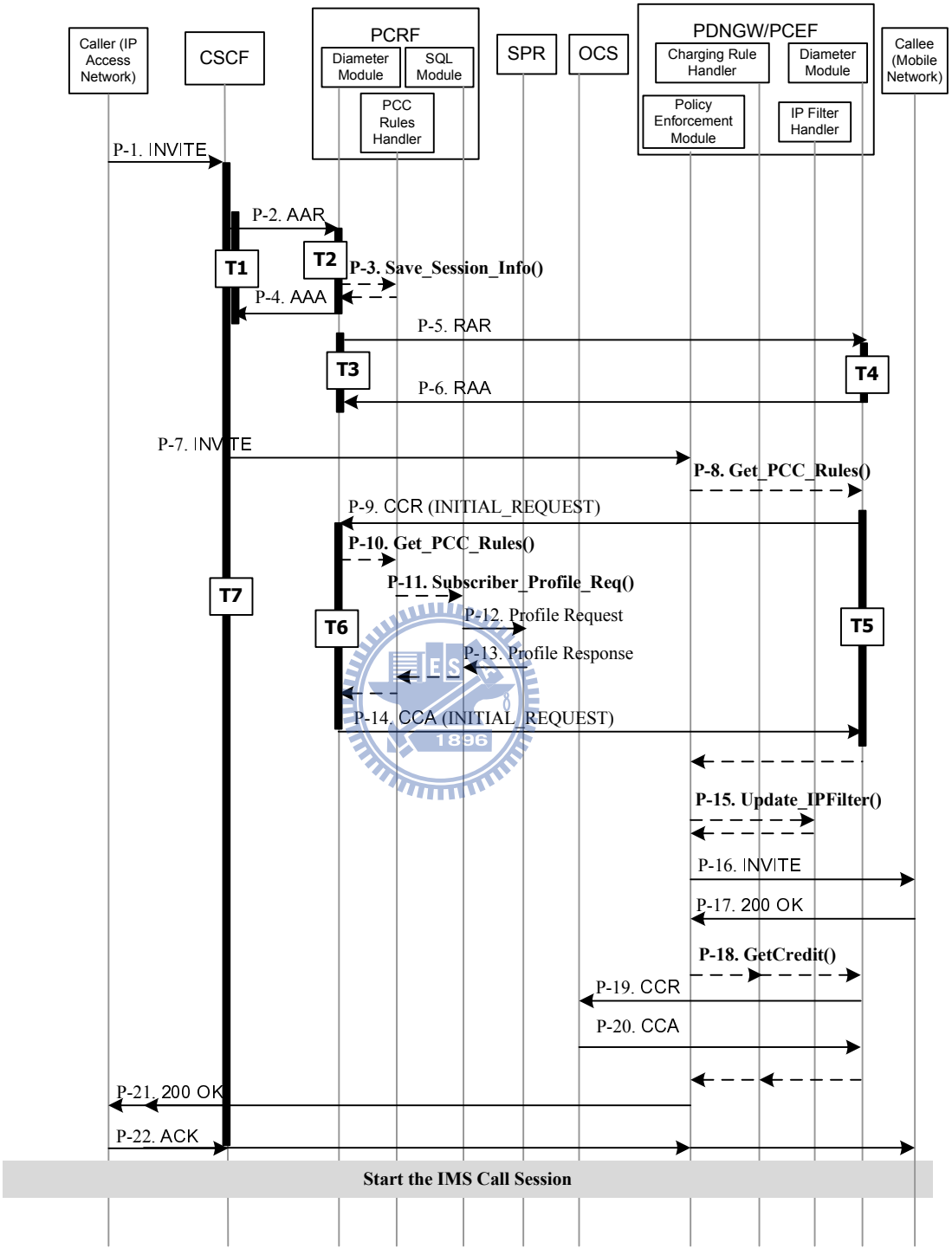


Figure 3.5: The Message Flow for the IMS Call Control Service in PCC

3.2.2 The Message Flow of the PCC Testbed

After the PCC rules have been configured in the SPR, a call session can be handled based on these policy settings. We use the IMS call control service as an example to show how our PCC testbed works. In the following scenario, a UE in the IP access network (Figure 3.2 (6)) attempts to make an IMS call to a mobile UE (Figure 3.2 (5)). The message flow for the IMS call control service is shown in Figure 3.5, and the details are given below.

Step P-1. The caller (i.e., a UE in the IP network) sends the SIP `INVITE` message to the CSCF to establish a new IMS call session. This message includes the media information in the Session Description Protocol (SDP) [31] specifying the subscriber ID and the service type.

Step P-2. When the CSCF receives the call session request, it saves the subscriber ID and the service type indicated in the SDP of the request message. Then it sends the Diameter `AAR` message with the session information (subscriber ID and service type) to the PCRF.

Step P-3. The PCRF Diameter Module invokes the **Save_Session_Info** function to store the session information.

Step P-4. When the session information is saved, the PCRF Diameter Module responds the Diameter `AAA` message to the CSCF to indicate that the traffic control is started.

Steps P-5 and P-6. The PCRF can optionally send the Diameter `Re-Auth Request (RAR)` message to the PDNGW/PCEF informing that the PCC rules (e.g., call duration or bandwidth control rules) need to be updated. The PDNGW/PCEF replies the Diameter `Re-Auth Answer (RAA)` message to the PDNGW/PCEF. The PCC rules re-authorization procedure is performed later in Steps P-9 to P-14.

Step P-7. The CSCF forwards the SIP `INVITE` to the PDNGW/PCEF.

Steps P-8 and P-9. The PCEF Policy Enforcement Module invokes the **Get_PCC_Rules** function, which instructs the Diameter Module to send the Diameter Gx `CCR` message that retrieves the PCC rules from the PCRF.

Steps P-10 and P-11. The PCRF Diameter Module invokes the **Get_PCC_Rules** that instructs the PCC Rules Handler to update the PCC rules. Then the PCRF PCC Rules Handler invokes **Subscriber_Profile_Req** based on the previously saved subscriber ID to request the subscriber profile which contains the PCC rules for the caller.

Step P-12. The SQL Module requests the subscriber profile (with the PCC rules) in the SPR.

Steps P-13 and P-14. The SPR returns the subscriber profile to the PCRF. Then the PCRF sends the Diameter Gx `CCA` message back to the PDNGW/PCEF. This `CCA` message contains the requested PCC rules.

Step P-15. According to the PCC rules of the subscriber profile, the PDNGW/PCEF Policy Enforcement Module invokes the **Update_IPFilter** function to update the iptables setting (call duration or bandwidth control) for packet filtering in the IP Filter Handler.

Step P-16. After the iptables has been configured, the PCEF Policy Enforcement Module continues to establish the IMS call by sending the SIP `INVITE` to the callee (i.e., a UE in the mobile network).

Step P-17. The callee acknowledges the PDNGW/PCEF with the SIP `200 OK`.

Step P-18. The Policy Enforcement Module invokes the **GetCredit** function that instructs the PCEF Diameter Module to retrieve the credit and check whether the credit is enough for this IMS call session.

Steps P-19 and P-20. The Charging Rules Handler requests credit from the OCS by exchanging the Diameter `CCR` and `CCA` messages. These two steps are the same as Steps D-1 and D-2 in Figure 1.3.

Step P-21. If the credit is enough, the Policy Enforcement Module forwards the `200 OK` to the caller.

Step P-22. The caller sends the SIP `ACK` message to indicate that the IMS call session is set up. The call session is established under the policy enforcement.

3.3 Performance Measurement

We implemented a prototype for the proposed PCC testbed. In our prototype, the PDNGW, the CSCF, the Web Server and the SPR are executed in the DELL OPTIPLEX 755 desktops with the Linux Fedora 8 operating system. The PCRF is executed in the Linux Fedora 8 environment in the ASUS F8S notebook. We use Windows messenger 5.1 as the SIP user agent function in our experiments. The components implemented in our testbed are described as follows:

- The Policy Enforcement Module and the CSCF are implemented based on the SIP server package OpenSIPS, version 1.5.
- The PDNGW and the PCRF are implemented and compiled by GNU C++ 4.1.2.
- The iptables package is used to exercise the IP Filter Handler for limiting the IP traffic in the PDNGW.
- The SPR is implemented by using MySQL 5.0.45 for storing subscription profile.

- The Web Server is supported by Apache 2.2.9 and we use PHP 5.2.6 packages for developing the web portal.
- The Diameter Modules in the CSCF, the PCRF and the PDNGW are implemented by OpenDiameter 1.0.7h.

Based on the above testbed, we performed 1000 IMS calls with PCC control. Figures 3.6-3.7 show the PCRF executing snapshots for the Gx CCR and CCA messages (i.e., P-9 and P-14 in Figure 3.5). The TFT-Packet-Filter-Information AVP in Figure 3.6 (1) presents the IP of the UE; therefore, the PCRF uses the IP to retrieve the UE's PCC rule in SPR. Then the PCRF installs the PCC rule (Figure 3.7 (1)-(3)). The Charging-Rule-Definition AVP defines the rule (Figure 3.7 (2)) to be installed. The Flow-Description AVP (Figure 3.7 (3)) determines the traffic for the service session.

Figure 3.8 shows the Diameter message packets in PCRF captured during an IMS call session. The IMS call setup procedure includes messages in Figure 3.8 (1) to Figure 3.8 (3). Figure 3.8 (1) presents the AAR and AAA messages of P-2 and P-4 in Figure 3.5, Figure 3.8 (2) presents the RAR and RAA messages of P-5 and P-6 in Figure 3.5, and Figure 3.8 (3) presents the CCR and CCA messages of P-9 and P-14 in Figure 3.5.

We also measure the message processing delay (i.e., T1 - T7) marked in Figure 3.5. The black bar indicates the message transmission period; for example, T1 starts from the message delivering time of the AAR in Step P-3 and ends at the receiving time of the AAA in Step P-5, while T2 starts from the receiving time of the AAR and ends at the delivering time of the AAA. The experimental data is shown in Table 3.1. We note that the overall message delay T7 is around 342ms. This period records the latency from the first SIP signaling message INVITE arrived at the CSCF in Step P-1 to the last message ACK in Step P-21. Our experiment data

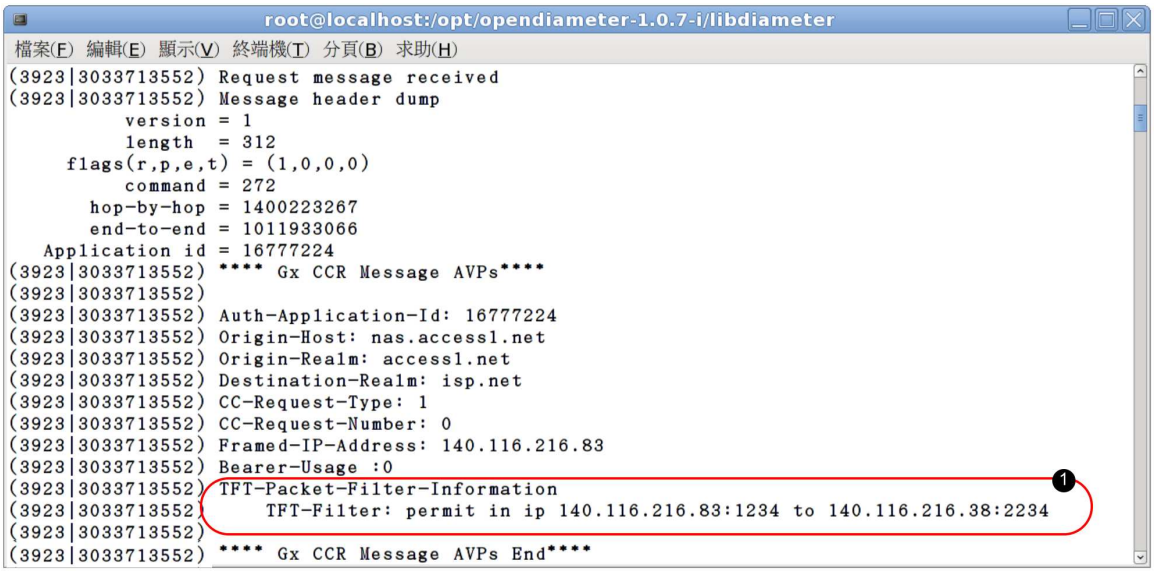


Figure 3.6: A Snapshot for the CCR Message Handling in Our Testbed

Table 3.1: The Message Data Signaling Delay Measured in Our Testbed

	T1	T2	T3	T4	T5	T6	T7
Mean (ms)	70.51	8.05	100.18	8.46	219.76	15.47	341.86
Variance (ms^2)	9278.73	57.41	13079.04	71.72	24013.61	145.96	24136.34
Deviation (ms)	96.33	7.58	114.36	8.47	154.96	12.08	155.36

indicates a large latency for PCC signaling in an IMS call setup.

3.4 Conclusion

In this chapter, we presented the design and the implementation of a testbed for the *Policy and Charging Control* (PCC) system proposed in 3GPP TS23.203. The advanced mobile services can be designed and supported through the PCC architecture, which brings operators extra revenue and increases the user interaction. Specifically, we used an IP multimedia call service example to demonstrate how to develop an advanced mobile service with policy and charging

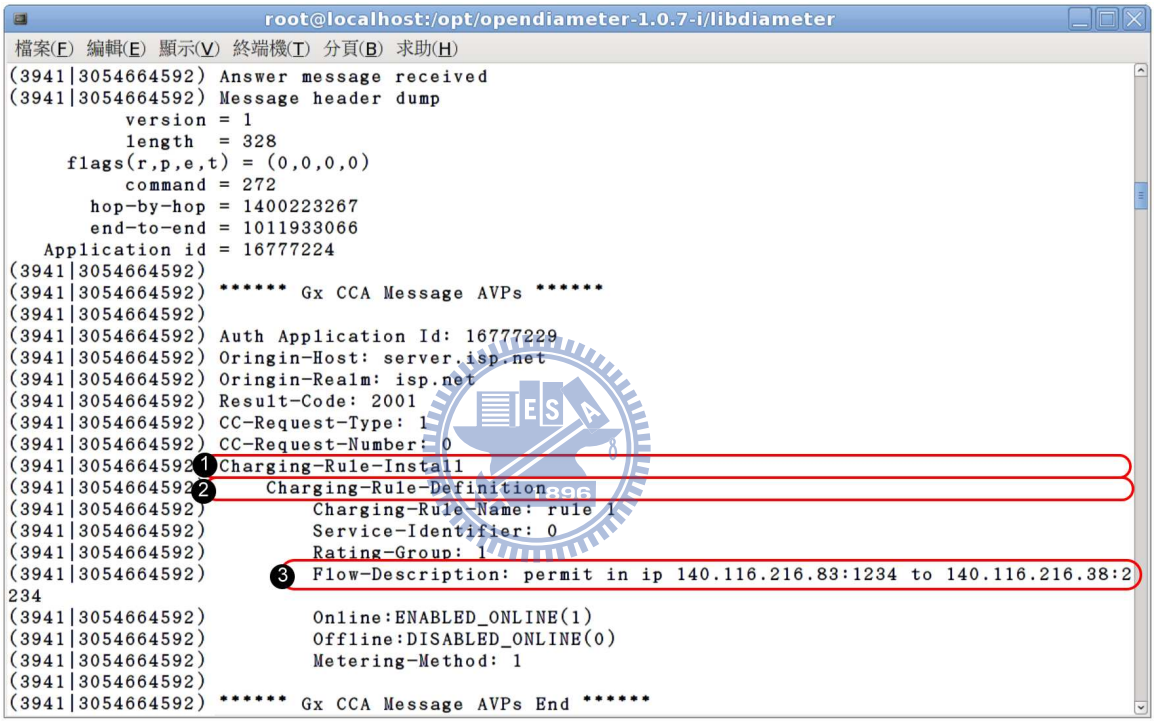


Figure 3.7: A Snapshot for the CCA Message Handling in Our Testbed

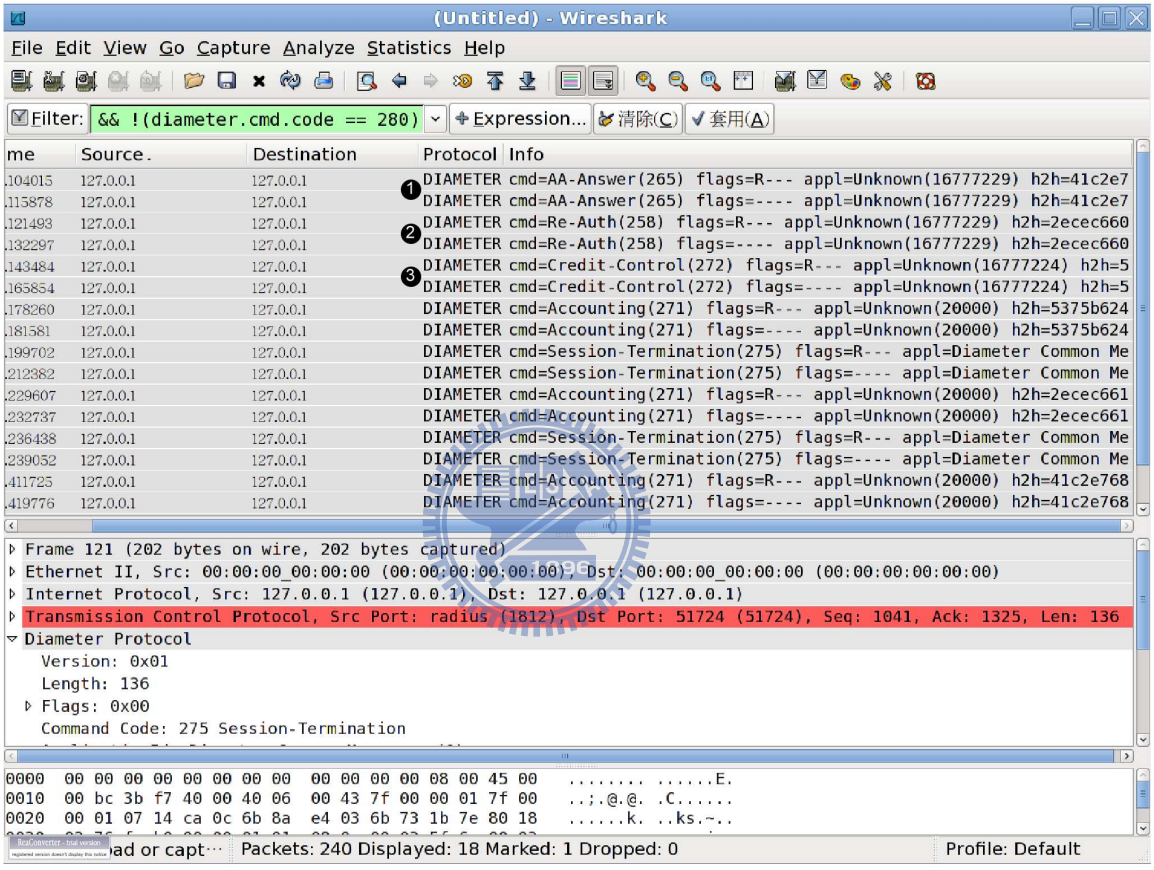


Figure 3.8: The List of Diameter Messages Captured in the PCRF

control in our testbed. We also provided the signaling delay measurement in this testbed. Finally, we observed that the PCC network delay is nonnegligible in an IMS call setup. Therefore, further works should be conducted to reduce signaling delay in the PCC system.

Chapter 4

Transparent Charging for IMS Services

Parlay is an efficient and flexible approach that enables telecom operators to efficiently wrap up their network services and capabilities and allows third parties to flexibly access those services for deployment of new applications that drive consumption of network services. Based on the PCC architecture described in the previous chapter, this chapter describes an Internet-mobile platform for telecom charging applications using *Parlay X*. We aggregate resources from the Internet and Next Generation Network (NGN) IP Multimedia Network Subsystem (IMS) mobile networks to enable "mashup" service creation. In our platform, we use the *IBM WebSphere software for Telecom* (WsT) to implement Parlay service capability that accommodates service oriented architecture services. The WsT is connected to the NGN/IMS platform for network capability provisioning. Then we use the Group Accounting System (GAS) as an example to illustrate how a new charging service can be created in the WsT platform and how the WsT interacts with the application server and NGN/IMS to provide GAS services.

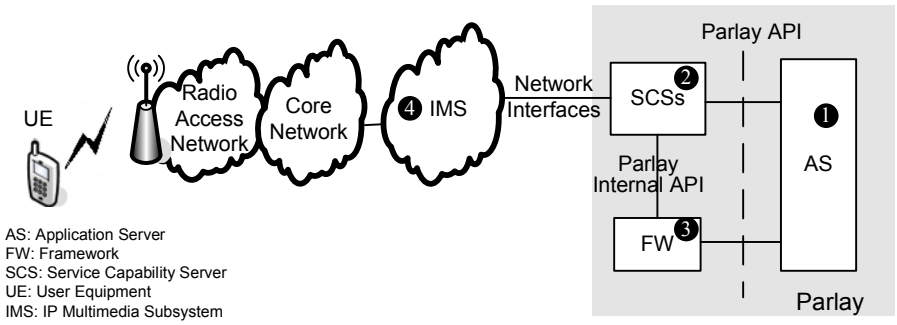


Figure 4.1: The Parlay Architecture

4.1 Parlay

In the Next Generation Network (NGN), the IP Multimedia Subsystem (IMS) integrates mobile telecom network with the Internet, which allows a telecom operator to flexibly bring attractive services to their customers [20, 16]. NGN/IMS utilizes *Session Initiation Protocol* (SIP) for signaling and Diameter for *Authentication, Authorization and Accounting* (AAA). On top of these protocols, Parlay can be implemented to make NGN/IMS service creation easier and more general through CORBA, WSDL or Java. The Parlay Group, together with the *European Telecommunications Standards Institute* (ETSI) and the *Third-Generation Partnership Project* (3GPP), has proposed the Parlay/*Open Service Access* (OSA) standard that defines application programming interfaces (APIs) for the control of mobile network capability [9, 27, 37, 28]. Furthermore, through Parlay, network capability is opened to third-party service providers other than the telecom operators to create and deploy new telecom services.

Figure 4.1 depicts the Parlay architecture. Parlay includes the *Application Server* (AS; Figure 4.1 (1)), the *Service Capability Servers* (SCSs; Figure 4.1 (2)) and the *Framework* (FW; Figure 4.1 (3)), and they offer an environment for application development, which is equivalent to the AS in Figure 1.1 (13). The SCSs provide the AS access to network capability functionali-

ties through the Parlay APIs. Before accessing an SCS, the AS should be authorized by the FW that provides an extensive framework of security and service management. Network access of third-party applications is subject to authentication and authorization. Parlay allows a telecom operator to set different privilege levels to the service providers according to the service level agreements (SLA). Some service providers are only allowed to receive notifications from the telecom network, while other highly-trusted providers can control calls and connections.

In 2003, Parlay Group defined a more complex and powerful web service of functionality exposed by Parlay API, namely Parlay X. This chapter describes an Internet-mobile platform for telecom applications based on Parlay X. Our platform aggregates resources of Internet and mobile networks to enable "mashup" service creation by using Parlay X APIs. Our solution uses the IBM WebSphere software for Telecom (WsT) to implement Parlay service capability that accommodates service oriented architecture (SOA) services. The WsT is connected to the NGN/IMS of Chunghwa Telecom (Figure 4.1 (4)), which is built on top of UMTS/LTE network, for network capability provisioning. Then we use "Group Accounting System (GAS)" as an example to illustrate how a new charging service can be created in the WsT platform and how the WsT interacts with the AS and NGN/IMS to provide GAS services.

4.2 IBM WebSphere software for Telecom

WsT (Figure 4.2 (3)) [1] is a platform providing NGN/IMS standard-compliant *network services*. A network service is a collection of self-contained functions that perform a defined task (e.g., call control and short message) and expose the task through a well-known interface [23]. Service providers only need to know what a network service does and what the interface is without knowing how the network service is implemented. The WsT consists of the following

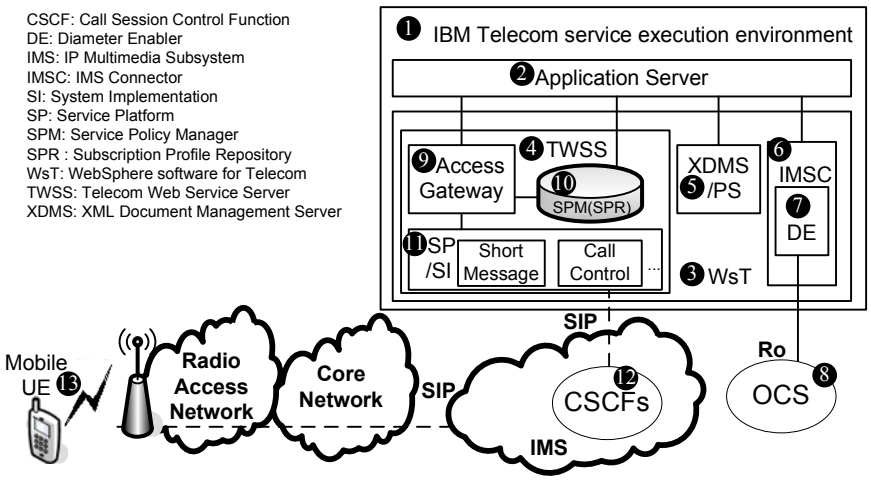


Figure 4.2: IBM WebSphere software for Telecom

products: *IBM WebSphere Telecom Web Services Server* (TWSS; Figure 4.2 (4)), *IBM WebSphere XML Document Management Server* (XDMS; Figure 4.2 (5)) and *IBM WebSphere IP Multimedia Subsystem Connector* (IMSC; Figure 4.2 (6)).

The TWSS enables service providers to access secure, reliable, and policy driven network services. Therefore, third-party service providers can enhance consumer and enterprise applications (resided in the AS; Figure 4.2 (2)) through open standard-based web services (e.g., Parlay X). The network services are located in *Web Service Implementations* (SI; Figure 4.2 (11)), which is connected to the *Call Session Control Function* (CSCF; Figure 4.2 (12)) in NGN/IMS network through SIP. The *Service Platform* (SP) provides common service implementation functions that enables more efficient and smaller deployment platform sharing among the network services. The *Access Gateway* (AG; Figure 4.2 (9)) plays the role of the Framework in Parlay. The AG provides a common control point for service providers to define, manage, and enforce policies and SLA for *requesters*. A requester can be either an application or a user equipment (UE; e.g., mobile UE; Figure 4.2 (13)). By using SLA, the TWSS determines

applications and users that can access certain network services under specific service policies. Such policies are stored in the *Service Policy Manager* (SPM; Figure 4.2 (10)), which provides management, storage, and retrieval functions for the policy rules (e.g., Quality of Service, such as minimum and maximum bit-rates). Every network service for a particular requester is associated with one or more policies. For example, call control service for different users and applications may have different charging policies (e.g., online/offline charging rate).

The XDMS provides storage, management and subscription to documents that are owned by entities within the IMS-based solution. In our solution, XDMS stores the buddy list of the group. The TWSS will use the Address List Management (ALM) web service to retrieve these buddy lists.

The IMSC is responsible for adapting the web services to the IMS protocols (i.e., Diameter and SIP). The adaptation of the Diameter protocol is implemented in the *Diameter Enabler* (DE; Figure 4.2 (7)). The DE allows the AS to retrieve and update NGN/IMS user data account, such as performing online charging through transactions with the *Online Charging System* (OCS; Figure 4.2 (8)). The DE receives online charging web service requests from the AS, sends an appropriate Diameter message (i.e., `Credit Control Request` (`CCR`)) to the OCS, receives authorization answers from the OCS (i.e., `Credit Control Answer` (`CCA`)), and sends the results back to the AS.

WsT is IBM's implementation for providing NGN network services. It not only provides the APIs for subscription and notification, but also provides Parlay X capability which accelerates the mobile application development by removing the need to learn details of the telecommunication network (e.g., IMS protocols). With IBM WsT capability, one can easily implement applications for next generation network.

Figure 4.3: The Graphical User Interface of iPhone for Group Accounting System

4.3 Service Provision for WsT-Based Group Accounting System

In this section, we show how the NGN/IMS provides network capability to the IBM WsT described in Section II. Moreover, we illustrate how to provide the network service to a service called Group Accounting System (GAS), which is based on the WsT.

Accounting is one of the most important business activities. GAS allows a group of people to share one or more accounts. An example of such group is the purchasing department of an enterprise with several on-going projects, with each project having its budget controlled by a project account. The GAS service is offered by WsT/IMS through a service access number, which has the same format as a telephone number. Through a UE's GAS application (Figure

Table 4.1: A Table Entry in the OCS's Account Database

User Name	Account Balance	Recharge Threshold
proj_a	50000	5000

Table 4.2: The GAS Entry in the CSCF Routing Table

User Telephone Number (Service Access Number)	Service Trigger Type	Initial Filter Criteria	
		Application Server	Method
+88621111111	MT	`WsT_ip`	`INVITE`

4.3), a member of this group can make a purchase charging from the project account. In Figure 4.3, the user purchases a $1000 USD laptop for *proj_a*. When the user clicks "Submit", the UE automatically dials the service access number, which results in a purchase request to the WsT.

The GAS interacts with the OCS to manage projects' account balances. These account balances are stored in an account database of the OCS. An entry of the simplified OCS account database is shown in Table 4.1. In this entry, the remaining credit of *proj_a* is $50000 USD. When the amount of the remaining credit in the OCS is less than a recharge threshold of $5000, the OCS will remind the project manager to refill this account [44]. A new purchase request can still be accepted until the remaining credit is depleted.

The provision of the GAS service through IMS consists of two parts. First, the GAS is enrolled in the WsT. Before the GAS is authorized to access the network service, we should set up a GAS policy rule in SPM. This policy rule sets GAS service as the requester to retrieve the Call Notification of the call control network service.

Second, the CSCF is configured to provide IMS network capabilities to the GAS. Specifically, we store the mapping of the GAS's service access number and WsT's IP address in the routing table of the CSCF as shown in Table 4.2. In this entry, the service access number of the GAS is +88621111111. The service type is *session terminating* or *mobile terminating* (MT;

which means that the service is triggered when the message is sent to the callee). The initial filter criteria include the IP address `WsT_ip` of the WsT and the SIP method that triggers this service (which is `INVITE` in this example).

4.4 Message Flows for Group Accounting System

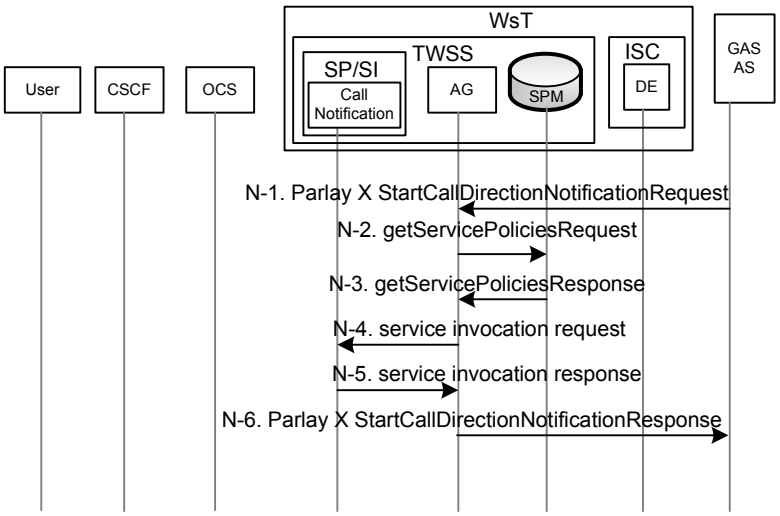
After the policy is configured in the SPM and the trigger profile is set up in the CSCF, the GAS AS can execute the notification process by initiating the "Call Notification" network service in the SP/SI of the WsT. Therefore, the GAS will be informed when a SIP message with the GAS's service access number is sent to the WsT. The notification message flow is shown in Figure 4.4 (a), and the details are given below.

Step N-1. The GAS AS sends the Parlay X `StartCallDirectionNotificationRequest` message to the AG to provide notification for its service access number. The Simple Object Access Protocol (SOAP) header of the message stores the requester (i.e., GAS) and the network service (i.e., Call Notification).

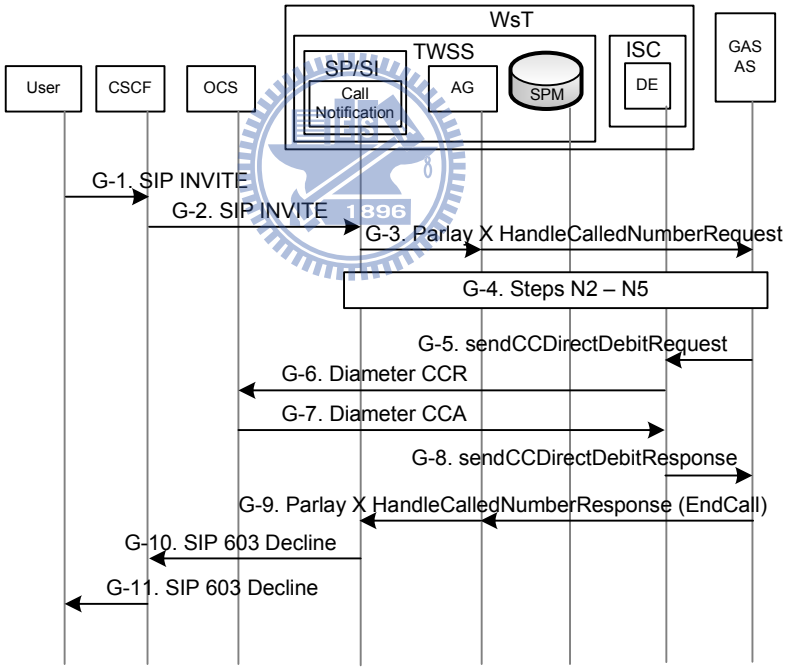
Step N-2. Upon receipt of the Parlay X message, the AG obtains the requester and the network service from the SOAP header.

Steps N-3 and N-4. The AG retrieves the policy rule from the SPM based on the requester "GAS" and the network service "Call Notification". The AG enforces the policy and performs SLA to authorize the offering of the GAS service through IMS.

Step N-5. The AG instructs the Call Notification network service to execute the notification operation.



(a) Notification Start



(b) Purchase

Figure 4.4: Message Flows for Group Accounting System

Steps N-6 and N-7. The Call Notification sends the response back to the GAS AS through `StartCallDirectionNotificationResponse` via the AG.

We note that Steps N-2 to N-5 are always executed whenever a Parlay X message arrives at the WsT. After the notification process, a GAS user can use a UE to request a purchase through NGN/IMS. Suppose that the user purchases a $1000 laptop for *proj a* through his/her UE. The message flow for this purchase is shown in Figure 4.4 (b), and the details are given below.

Step G-1. After the user clicks the "Submit" button (see Figure 4.3), the UE automatically dials the GAS's service access number +88621111111, which results in a SIP `INVITE` message delivered to the CSCF. This message includes the purchase information: the price ($1000), the item name (laptop), the project ID (*proj_a*) and the purchase/receipt dates (optional). The information is carried in the display-name in the From header of the `INVITE` message.

Step G-2. By retrieving +88621111111 in the To header of the SIP message, the CSCF checks if the service access number matches the MT filter criteria of the trigger profile. If the service trigger is matched, the IP address `WsT_ip` is used to route the `INVITE` message to the WsT.

Steps G-3 and G-4. When the WsT SP/SI receives the SIP `INVITE` message with GAS's service access number +88621111111, the WsT notifies the GAS AS through the Parlay X `HandleCalledNumberRequest`. In this Parlay X message, the parameter CallingParticipantName carries the purchase information copied from the display-name of the SIP `INVITE`. Then Steps N-2 to N-5 are executed for policy enforcement.

Step G-5. The GAS AS retrieves the purchase information ($1000, laptop and *proj_a*) in the CallingParticipantName parameter, and invokes `sendCCDirectDebitRequest`, which

instructs the WsT to request the credit ($1000 USD) through the DE. Note that the interface between DE and AS is not defined in Parlay X. This interface follows SOAP specified by IBM.

Step G-6. Accordingly, the DE sends a Diameter `CCR` message to the OCS to request credit from *proj_a*'s account. This message is the same as Step D-1 in Figure 1.3.

Step G-7. The OCS deducts $1000 from *proj_a*'s account. After the deduction, the OCS checks if the new balance ($49000) is more than the recharge threshold. If not, the OCS will remind the project manager to refill the project account by, e.g., sending a short message. Then the OCS sends a Diameter `CCA` message back to DE. The message is the same as Step D-2 in Figure 1.3, and it indicates that the credit deduction is successfully preformed.

Step G-8. The DE sends the SOAP `sendCCDirectDebitResponse` message to the GAS AS to indicate that the credit request is successful.

Step G-9. The GAS AS returns the Parlay X `HandleCalledNumberResponse` message to the WsT to inform the user that the purchase is approved. Since the call is not necessary to be answered, we simply end the call by setting the ActionToPerform parameter of the message to "EndCall" to reject the call.

Steps G-10 and G-11. The SP/SI sends a SIP `603 Decline` message to the UE through the CSCF. The purchase is successfully completed.

4.5 Application Development

Based on the message flows described in the previous section, this section shows how to utilize the ParlayX API to implement our application. We use IBM Rational Application Developer

```
        try {
            // Get the Service Endpoint
(1)         CallDirectionManagerProxy proxy = new CallDirectionManagerProxy();
(2)         proxy.setEndpoint("http://localhost:9080/ParlayX21Web/services/CallNotification");

            // Set notification endpoint definition (callback reference)
(3)         SimpleReference ref = new SimpleReference();
(4)         ref.setCorrelator("cor_gas");
(5)         ref.setEndpoint(new java.net.URI
                ("http://localhost:8111/CallNotification/services/CallNotification?WSDL"));

            // Set address of terminal
(6)         String uriArray [] = "+886211111111";

            // Create array of notification events
(7)         CallEvents[] events = new CallEvents[1];
(8)         events[0] = CallEvents.CalledNumber;

            // Execute the web service
(9)         proxy.startCallDirectionNotification(ref, uriArray, events);
(10)    } catch (Exception e) {
(11)        // exception handling
        }
```

Figure 4.5: StartCallDirectionNotificationRequest Web Service Program Segment

with WebSphere Telecom Toolkit feature to develop GAS. The Rational Application Developer is an integrated development environment (IDE) that provides a development platform to design, develop, test and deploy web service applications. In our current situation, since we are still proceeding the WsT/Chunghwa Telecom integration, we use the WebSphere Telecom Toolkit's simulator which simulates real telecommunication network for functional testing.

We use the Parlay X `StartCallDirectionNotificationRequest` as an example to illustrate how to access a web service service. Figure 4.5 shows the program segment for executing the call notification web service. The proxy object in Line 1 is responsible for requesting service, and its endpoint (i.e., the access point of a web service) is set to WebSphere Telecom Toolkit simulator's call notification web service Uniform Resource Identifier (URI) in Line 2. However, once the WsT/Chunghwa Telecom integration is done, we will connect the GAS to the integrated WsT environment by simply modifying the endpoint to the WsT's call notification URI. Lines 3 - 5 set the notification endpoint (i.e., the callback reference) to the GAS's URI. That is, the GAS will be informed through the notification endpoint. Line 6 sets up the address that the program is going to monitor as the service access number of the GAS (+88621111111). Lines 7 and 8 configure the notification event type as CallEvents.CalledNumber; therefore, calls to the number (+88621111111) will be handled by our program. After all the settings, we execute the web service by invoking the `startCallDirectionNotification` on the proxy object. If any error occurs, we will handle the exception in Line 11.

4.6 Conclusion

In this chapter, we presented how to integrate the IBM WebSphere software for Telecom (WsT) with the NGN/IMS platform for network capability provision. This WsT/IMS integration ac-

commodates flexible service oriented architecture (SOA) services. Through the standard-based web service APIs offered by the WsT, application developers can create new services without knowing the details of the telecommunication network. We use Group Accounting System (GAS) as an example to illustrate how a new charging application can be easily created in the WsT platform. Specifically, we show how the WsT interacts with the AS and NGN/IMS to provide the GAS service.

Chapter 5

Conclusions and Future Work

In the *Universal Mobile Telecommunications System* (UMTS), the *IP Multimedia Subsystem* (IMS) is developed to provide multimedia services. In order to successfully promote IMS services, charging has become a major concern of operators. Through online charging, an operator can ensure that credit limits are enforced and resources are authorized on a per-transaction basis. In this dissertation, we first studied an online credit reservation procedure for prepaid users. Then we presented the design and the implementation of a testbed for the policy and charging control system and how to provide application-level charging service. This chapter concludes our work presented in this dissertation, and briefly discusses future directions of our work.

5.1 Concluding Remarks

Chapter 2 and 3 discuss charging issues in the network level. In Chapter 2, we studied the online credit reservation procedure for prepaid users in the online charging system. If the assigned credit units are consumed before the session is completed, a reserve units operation is executed to obtain more credit units from the OCS. During the RU operation, packet delivery is suspended until extra credit units are granted from the OCS. To avoid session suspension during

credit reservation, we proposed the credit pre-reservation mechanism that reserves credit earlier before the credit at the network is actually depleted. Analytic and simulation models were developed to investigate the performance of this credit pre-reservation mechanism (CPM). Our study provided guidelines to set up the parameters for our proposed mechanism.

In Chapter 3, we presented a testbed to investigate the impact of PCC on the advanced mobile service deployment. Specifically, we used the IMS call service as an example to demonstrate how to implement an advanced mobile service with PCC in our testbed. This IMS call service provided web-based user interface for an administrator to conveniently set up the service admission policy. These policy settings were converted to PCC rules and were used to ensure the QoS. This testbed allowed operators to experiment on advanced mobile services with PCC control without implementing these services in the commercial telecom system. We also provided measurement results to investigate the performance of the PCC testbed.

Chapter 4 focused on an application level service platform, where the IMS service-oriented architecture (SOA) services can be development through Parlay X. Through the standard-based web service APIs furnished by the service platform, application developers can create services without knowing the details of the telecommunication network such as PCC protocols. We used Group Accounting System (GAS) as an example to illustrate how a new charging application can be easily created and provided in our service platform.

5.2 Future Work

Based on the research results of this dissertation, we suggest the following topics for further study.

Reducing Signaling Delay in the PCC System: Chapter 2 provided the signaling delay measurement in this testbed. The measurement results indicate that the PCC network delay is unnegligible in an IMS call setup. Moreover, in the architecture defined in 3GPP 23.203 [21] and 23.402 [18], enforcing PCC rules in 3GPP and non-3GPP interconnected networks involves more signaling messages (especially, handover between 3GPP and non-3GPP networks) and therefore causes longer network delay. We will analyze the handover performance in heterogeneous networks, and then devise a mechanism for reducing signaling delay.

IBM WsT and Chunghwa Telecom IMS Integration: In our current implementation, the charging application is connected to IBM Rational Application Developer Telecom Toolkit, which provides a simulation environment for Parlay and IMS network. We will integrate the IBM WsT and the IMS of Chunghwa Telecom as the application service platform for a real-world deployment. Once the integration is done, we will connect the GAS described in Chapter 4 to the integrated WsT environment by simply modifying the endpoint to the WsT's URI.

NGN/IMS Application Development: We described the Internet-mobile platform for telecom applications in Chapter 4. Our platform provides an environment to enable "mashup" service creation. For example, we can build a social network with telecom communication capabilities, namely "JoinMe" service. With this service, a mobile user can announce an immediate activity to nearby associates with common sets of subscriptions via the web page, voice phone call, short message service (SMS) and multimedia messaging service (MMS) [11], and the service can be charged through our platform. The published activity will target a set of subscribers that are unknown to the initiator of the activity. The intelligent application will match the activity to individuals who subscribe to common topics

with the activity. Any subscriber who wants to join in the activity can reply through the same methods (e.g., voice, SMS, MMS). The mobile and instantaneous nature of the platform will help us creating new values and offering differentiated NGN/IMS services on the Internet.

The Performance for Credit Pre-reservation Mechanism: In this dissertation, we proposed an analytic model to investigate the CPM performance for exponential distribution packet interarrival time. This analytic model adopted exponentially-distributed packet interarrival assumptions, and the conclusion under such assumptions may be misleading in the presence of heavy-tailed distributions (e.g., Pareto distribution). In the future, we will consider the Pareto distribution, which can accurately reflect the nature of heavy-tail internet traffic in the multiple session telecommunication networks. It has been shown that the Pareto distribution with the shape parameter $1 \leq b \leq 2$ can approximate the packet traffic very well. Modelling of heavy-tail traffic is necessary so that networks can be provisioned based on accurate assumptions of the traffic that they carry. Therefore, we will extend the packet interarrival time in our analytic models for Pareto distribution.

Bibliography

[1] IBM WebSphere software for Telecom, Second Edition. http://publib.boulder.ibm.com/infocenter/wtelecom/v7r0m0/index.jsp.

[2] IMS DIAMETER Toolkit. RADVISION. 2007.

[3] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Architectural Requirements for Release 1999 (Release 1999). Technical Specification 3G TS 23.121 Version 3.6.0 (2002-06), 2002.

[4] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; Telecommunication Management; Charging Management; Charging data description for the Packet Switched (PS) domain. Technical Specification 3G TS 32.215 version 5.9.0 (2005-06), 2005.

[5] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Charging rule provisioning over Gx interface. Technical Specification 3G TS 29.210 Version 6.7.0 (2006-12), 2006.

[6] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Telecommunication management; Charging management; Charging data

description for the IP Multimedia Subsystem (IMS). 3rd Generation Partnership Project . Technical Specification 3G TS 32.225 Version 5.11.0 (2006-03), 2006.

[7] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Policy control over Gq interface. Technical Specification 3G TS 29.209 Version 6.7.0 (2007-06), 2007.

[8] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Rx Interface and Rx/Gx signalling flows. Technical Specification 3G TS 29.211 Version 6.4.0 (2007-06), 2007.

[9] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Open Service Access (OSA); Parlay X Web Services; Part 1: Common (Release 9). Technical Specification 3G TS 29.199-01 Version 9.0.0 (2009-12), 2009.

[10] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; End-to-end Quality of Service (QoS) concept and architecture. Technical Specification 3G TS 23.207 version 9.0.0 (2009-12), 2009.

[11] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Messaging Service (MMS); Stage 1. 3rd Generation Partnership Project . Technical Specification 3G TS 22.140 Version 9.0.0 (2009-12), 2009.

[12] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3. 3G TS 24.229 Version 10.0.0 (2010-06), 2010.

[13] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Core Network; IP Multimedia (IM) session handling; IM call model; Stage 2. Technical Specification 3G TS 23.218 Version 10.0.0 (2010-06), 2010.

[14] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Telecommunication management; Charging management; Charging architecture and principles. Technical Specification 3G TS 32.240 Version 9.1.0 (2010-6), 2010.

[15] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Telecommunication management; Charging management; Diameter charging applications. Technical Specification 3G TS 32.299 Version 9.4.0 (2010-06), 2010.

[16] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Telecommunication management; Charging Management; IP Multimedia Subsystem (IMS) charging. Technical Specification 3G TS 32.260 Version 10.0.0 (2010-06), 2010.

[17] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Telecommunication management; Charging management; Online Charging System (OCS): Applications and interfaces. Technical Specification 3G TS 32.296 Version 10.0.0 (2010-04), 2010.

[18] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Architecture enhancements for non-3GPP accesses. Technical Specification 3G TS 23.402 Version 10.0.0 (2010-06), 2010.

[19] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access. Technical Specification 3G TS 23.401 Version 10.0.0 (2010-06), 2010.

[20] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2. Technical Specification 3G TS 23.228 Version 10.1.0 (2010-06), 2010.

[21] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Policy and charging control architecture. Technical Specification 3G TS 23.203 Version 10.0.0 (2010-06), 2010.

[22] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service Description; Stage 2. Technical Specification 3G TS 23.060 Version 10.0.0 (2010-06), 2010.

[23] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Terminals; Technical realization of the Short Message Service (SMS). 3rd Generation Partnership Project . Technical Specification 3G TS 23.040 Version 9.2.0 (2010-03), 2010.

[24] Calhoun, P., Loughney, J., Guttman, E., Zorn, G. and Arkko, J. Diameter Base Protocol. IETF RFC 3588, September 2003.

[25] Chang, M.-F. and Yang, W.-Z. Performance of Mobile Prepaid and Priority Call Services. *IEEE Communications Letters*, 6(2):61–63, 2002.

[26] Cheng, M. and Chang, L.-F. Wireless Dynamic Channel Assignment Performance under Packet Data Traffic. *IEEE Journal on Selected Areas in Communications*, 17(7):1257–1269, July 1999.

[27] Chlamtac, I., Lee, H.-Y., Lin, Y.-B., and Tsai, M.-H. An OSA Service Capability Server for Mobile Services. *International Journal of Pervasive Computing and Communications*, 4(3), 2008.

[28] Chou, C.-M., Hsu, S.-F., Lee, H.-Y., Lin, Y.-C., Lin, Y.-B., and Yang, R.-S. CCL OSA: A CORBA-based Open Service Access System. *International Journal of Wireless and Mobile Computing*, 1(3-4):289–295, 2006.

[29] Collins, D. *Carrier Grade Voice over IP*. McGraw-Hill, 2003.

[30] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M. and Loughney, J. Diameter Credit-Control Application. IETF RFC 4006, August 2005.

[31] Handley, M., Jacobson, B., and Perkins, C. SDP: Session Description Protocol. IETF RFC 4566, July 2006.

[32] Lau, J. and Liang, B. Optimal Pricing for Selfish Users and Prefetching in Heterogeneous Wireless Networks. *IEEE International Conference on Communications*, pages 24–28, June 2007.

[33] Lee, H.-Y., and Lin, Y.-B. A Cache Scheme for Femtocell Reselection. *IEEE Communications Letters*, 14(1):27–29, January 2010.

[34] Lin, P., Chen, H.-Y., Fang, Y., Jeng, J.-Y., and Lu, F.-S. A Secure Mobile Electronic Payment Architecture Platform for Wireless Mobile Networks. *IEEE Transactions on Wireless Communications*, 7(7):2705–2713, 2008.

[35] Lin, Y.-B., and Pang, A.-C. *Wireless and Mobile All-IP Networks*. Wiley, 2005.

[36] Lin, Y.-B., and Sou, S.-I. *Charging for Mobile All-IP Telecommunications*. John Wiley & Sons, 2008.

[37] Lofthouse, H., Yates, M. J. and Stretch R. Parlay X Web Services. *BT Technology Journal*, 22(1):1358–3948, 2004.

[38] Open Diameter Project. Open Diameter 1.0.7 h. http://www.opendiameter.org/.

[39] openSIPS Project. http://www.opensips.org/.

[40] Pang, A.-C., and Chen, Y.-K. A Multicast Mechanism for Mobile Multimedia Messaging Service. *IEEE Transactions on Vehicular Technology*, 53(6):1891–1902, November 2004.

[41] Rigney, C., Willens, S., Rubens, A. and Simpson W. Remote Authentication Dial In User Service (RADIUS). IETF RFC 2865, June 2000.

[42] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E. SIP: Session Initiation Protocol. IETF RFC 3261, June 2002.

[43] Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V. RTP: A Transport Protocol for Real-Time Applications. IETF RFC 1889, January 1996.

[44] Sou, S.-I., Hung, H.-N., Lin, Y.-B., Peng, N.-F., and Jeng, J.-Y. Modeling Credit Reservation Procedure for UMTS Online Charging System. *IEEE Transactions on Wireless Communications*, 6(11):4129–4135, 2007.

[45] Sou, S.-I., Hung, Y.-B., Lin, Y.-B., Peng, N.-F., and Jeng, J.-Y. Modeling Credit Reservation Procedure for UMTS Online Charging System. *IEEE Transactions on Wireless Communications*, 6(11):4129–4135, 2007.

[46] Sou, S.-I., Lin, Y.-B., and Jeng, J.-Y. Reducing Credit Re-authorization Cost in UMTS Online Charging System. *IEEE Transactions on Wireless Communications*, 7(9):3629–3635, 2008.

[47] Sou, S.-I., Lin, Y.-B., Wu, Q. and Jeng, J.-Y. Modeling Prepaid Application Server of VoIP and Messaging Services for UMTS. *IEEE Transactions on Vehicular Technology*, 56(3):1434–1441, May 2007.

[48] The Netfilter/Iptables Project. http://www.netfilter.org/.

[49] Watson, E.J. *Laplace Transforms And Applications*. Birkhauserk, 1981.

[50] Wu, D., Hou, T., and Zhang, Y.-Q. Transporting Real-time Video over the Internet: Challenges and Approaches. *Proceedings of the IEEE*, 88(12):1855–1875, December 2000.

[51] Yang, S.-R., Lin, Y.-B. Performance Evaluation of Location Management in UMTS. *IEEE Transactions on Vehicular Technology*, 52(6):1603–1615, November 2003.

Appendix A

The Simulation Model for Credit Pre-reservation Mechanism

This appendix describes the discrete event simulation model for the *credit pre-reservation mechanism* (CPM). The simulation defines four types of events to represent a new session arrival (S_arrival), a packet arrival for a session (P_arrival), a `CCR` message delivery (CCR), and a `CCA` message delivery (CCA). Both the CCR and the CCA have three sub-types: "INITIAL_REQUEST", "UPDATE_REQUEST", and "TERMINATE_REQUEST". In a simulation run, the events are inserted into an event list and are removed (and processed) from the list in the increasing timestamp order.

An event e has a field $e.s$ that represents the session of this event. Several variables are used to calculate the output measures B, W, P_r, P_{nc}, and X_s in a simulation run, including:

- n_b: the number of buffered packets
- n_l: the number of the LC periods where more than one RU operation is executed
- n_p: the number of packet arrivals

- n_r: the number of RU operations executed
- n_s: the number of session arrivals
- N_b: the number of blocked session requests due to insufficient credit units
- N_f: the number of force-terminated sessions due to insufficient credit units
- N_l: the number of LC periods
- T: the summation of the waiting times for all buffered packets

From the above variables, we compute the following output measures:

$$B = \frac{n_b}{n_r}, W = \frac{T}{n_p}, P_r = \frac{n_l}{N_l}, P_{nc} = \frac{N_b + N_f}{n_s}, \text{ and } X_s = \frac{n_r}{n_s}$$

The following variables are also used in the simulation:

- C_r: the amount of the remaining credit in a user account at the OCS
- N_a: the number of currently active sessions
- $s.c$: the amount of remaining allocated credit units at session s
- $s.n_b$: the number of currently buffered packets at session s

Figure A.1 illustrates the flowchart of the simulation model (Steps 1-21). It contains two modules: the CCR and the CCA modules shown in Figures A.2 (Steps 22-30) and A.3 (Steps 31-44), respectively. The simulation terminates when the total credit depletes (i.e., when $C_r < \theta$) and all existing sessions are completed. The details are described as follows:

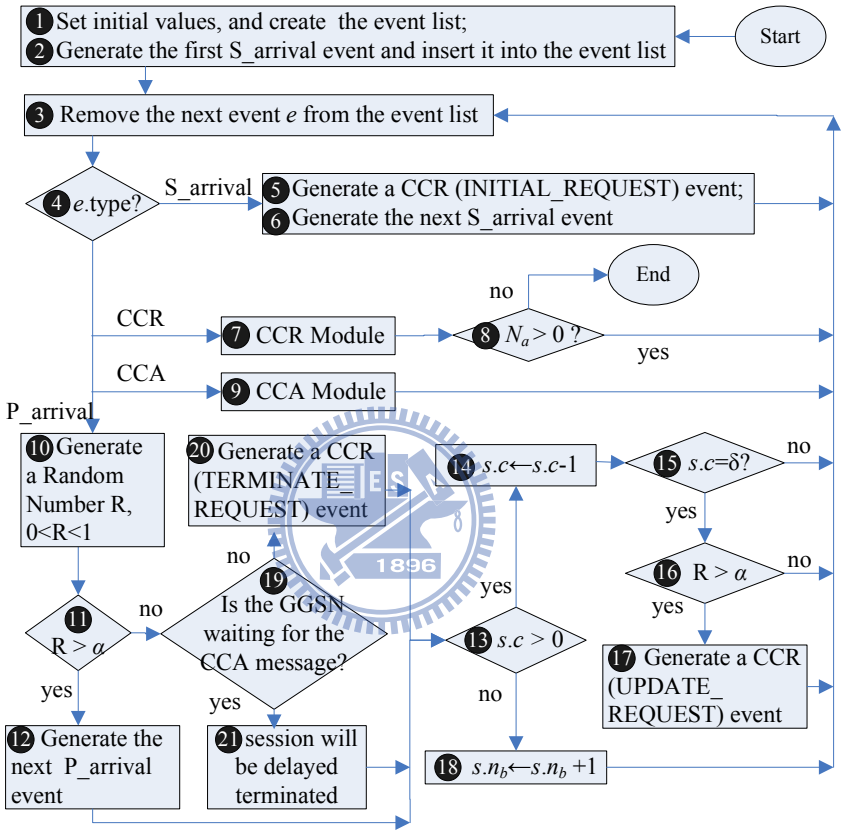


Figure A.1: Flowchart of the simulation model for the CPM

Steps 1-2. The initial values (e.g., C_r, N_a, N_s, etc.) are set up. The event list is set empty. The first S_arrival event is generated and inserted into the event list.

Steps 3-4. The next event e is removed from the event list. This event is executed based on its event type.

Steps 5-6. [e.type=S_arrival] A CCR (INITIAL_REQUEST) event and the next S_arrival event are generated and inserted into the event list. The N_s value is increased by one. The simulation proceeds to Step 3.

Step 7. [e.type=CCR] The CCR module in Figure A.2 is executed. This module is used to process the `CCR` message (see Steps 22-30).

Step 8. If there is no active sessions (i.e., $N_a = 0$; which implies that $C_r < \theta$ and the event list is empty), the statistics (B, W, P_r, P_{nc} and X_s) are calculated, and the simulation terminates. Otherwise, the simulation proceeds to Step 3.

Step 9. [e.type=CCA] The CCA module in Figure A.3 is executed. This module is used to process the `CCA` message (see Steps 31-44).

Steps 10-11. [e.type=P_arrival] N_p is increased by one. The simulation generates a random number R (where $0 < R < 1$) to determine whether the arrival packet is the last one or not. If $R > \alpha$ (the packet is not the last one), Step 12 is executed. Otherwise, Step 19 is executed.

Step 12. The next P_arrival event for session $e.s$ is generated and inserted into the event list according to the increasing timestamp order.

Step 13. If s has some credit units (i.e., $s.c > 0$), Step 14 is executed. Otherwise, Step 18 is executed.

Step 14. The $s.c$ value is decreased by one (i.e., the arrived packet is successfully processed).

Step 15. If $s.c$ equal to the threshold value δ (i.e., $s.c = \delta$), Step 16 is executed to reserved more credit units. Otherwise, Step 3 is executed.

Step 16. If $R > \alpha$, and if the previous `CCR` message has successfully reserved credit, Step 17 is executed. Otherwise (session is terminated), there is no need to send the `CCR` message, and the simulation directly proceeds to Step 3.

Step 17. A CCR(UPDATE_REQUEST) event is generated to request for more credit units. The simulation proceeds to Step 3.

Step 18. If $s.c = 0$, at Step 13, the arrival packet is buffered and the $s.n_p$ value is increased by one. The simulation proceeds to Step 3.

Steps 19-21. At Step 11, if the arrived packet is the last one that arrives during the RU operation, the CCR(TERMINATE_REQUEST) is delayed delivered after the GGSN has received the `CCA` message (which occurs at Step 36 in Figure A.3). Otherwise, Step 20 generates a CCR(TERMINATE_REQUEST) event and inserts the event into the event list. Then the simulation proceeds to Step 13.

Steps 22-29 describe the CCR module.

Step 22. The sub-type of the CCR event is checked and is executed.

Steps 23-25. [subtype=INITIAL_REQUEST] If the OCS has enough credit units (i.e., $C_r \geq \theta$), the C_r value is decreased by θ, the number of the active sessions N_a is increased by one, and the CCA(INITIAL_REQUEST) event is generated and inserted into the event

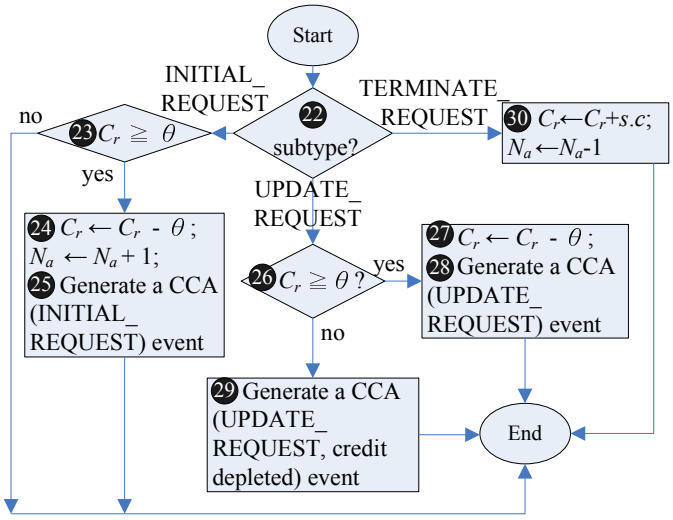


Figure A.2: Flowchart of the CCR module

list. Then the simulation returns to Step 8 in Figure A.1. If $C_r < \theta$ at Step 23, then the new session is rejected, and the CCR module exits without taking any action.

Steps 26-29. [subtype=UPDATE_REQUEST] If $C_r \geq \theta$, then C_r is decreased by the amount θ, and the CCA(UPDATE_REQUEST) event is generated and inserted into the event list. Otherwise, a CCA(UPDATE_REQUEST, credit depleted) event is generated and inserted into the event list (i.e. the simulation rejects the `CCR`). The simulation returns to Step 8 in Figure A.1.

Step 30. [subtype=TERMINATE_REQUEST] The session s refunds the credits to the OCS (i.e., $C_r \leftarrow C_r + s.c$). The number of the active sessions N_a is decreased by one.

Steps 31-43 describe the CCA module.

Step 31. The sub-type of the CCA event is checked and executed.

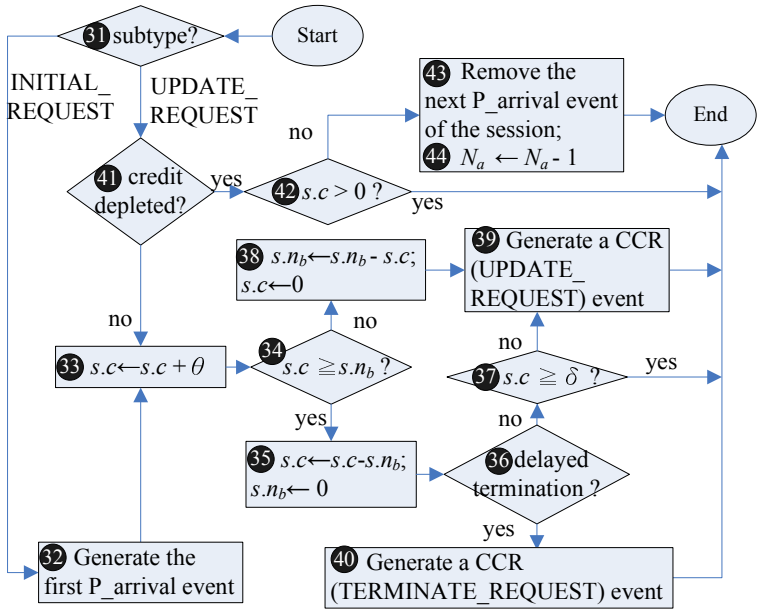


Figure A.3: Flowchart of the CCA module

Step 32. **[subtype=INITIAL_REQUEST]** The first P_arrival event e (with the current timestamp) of session $e.s$ is generated and inserted into the event list.

Step 33. The amount θ of credit units are reserved for session s (i.e., the $s.c$ value is increased by θ).

Step 34. The simulation checks whether the credit of session s suffices to deliver the buffered packets (i.e., $s.c \geq s.n_b$). If so, Step 35 is executed. Otherwise, Step 38 is executed.

Steps 35-36. **[$s.c \geq s.n_b$]** The $s.c$ value is decreased by $s.n_b$, and $s.n_b$ is set to 0. If s is delayed termination (Step 21 in Figure A.1), Step 40 is executed. Otherwise, Step 37 is executed.

Step 37. If $s.c < \delta$, Step 39 is executed. Otherwise, the simulation returns to Step 3 in Figure A.1.

Step 38. **[$s.c < s.n_b$]** The $s.n_b$ value is decreased by $s.n_b$, and $s.n_b$ is set to 0.

Step 39. [$s.c < \theta$] A CCR(UPDATE_REQUEST) event is generated and inserted into the event list.

Step 40. A CCR(TERMINATE_REQUEST) event is generated and inserted into the event list.

Step 41. **[subtype=UPDATE_REQUEST]** If the CCR event indicates that the credit is depleted (Step 29 in Figure A.2), Step 42 is executed. Otherwise, Step 33 is executed.

Steps 42-44. If $s.c > 0$, the GGSN will continue to deliver the packets until $s.c = 0$ or s is terminated. Otherwise ($s.c = 0$), the next P_arrival event of s in the event list is removed, the session is terminated, and N_a is decreased by one.

Appendix B

The Implementation of the PCC Diameter Modules

To implement the PCC Diameter Modules described in Chapter 3, we modified the Open Diameter API [38] for diameter messages. Open Diameter API is a session-based API, which realizes the diameter protocol defined in RFC 3588 [24]. In this API, a diameter session is represented by a C++ class. The session classes can be viewed as client or server, which provide AAA client and AAA server capabilities, respectively. These two types of classes are further sub-divided into authentication/authorization session classes and accounting session classes. All of these session classes are derived from a specific AAA state machine as defined in [24].

In Open Diameter framework, client application should create an instance of the client session class, for example, AAA_ClientAuthSession (Figure B.1 (1)), which represents the authentication/authorization session, or classes derived from it. AAA_ClientAuthSession provides virtual functions that are called when incoming session message is received. Classes derived from AAA_ClientAuthSession may overwrite the virtual functions to provide specific functionality. As shown in Figure B.1, AAA_ClientAuthSession derives from AAA_SampleAuthClient (Figure B.1 (2)), which is originally from the open diameter distribution, and we use it to implement

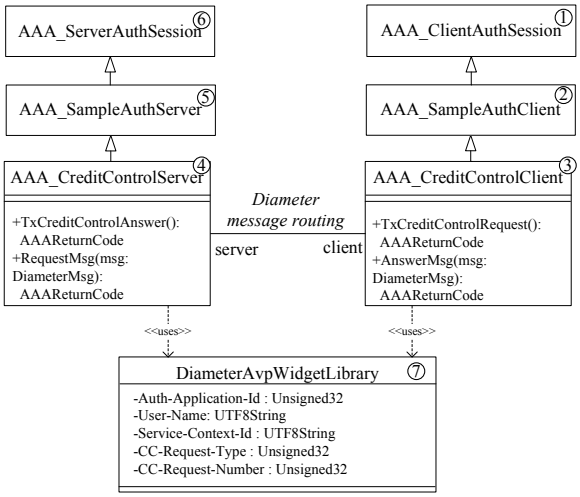


Figure B.1: Class Diagram of Our Purposed Open Diameter Application

an example diameter client session. Since in the current Open Diameter version, it only realizes the protocol in RFC 3588 and does not support messages in other specifications (e.g., the diameter Gx `CCR/CCA` messages in RFC 4006), we reuse AAA_SampleAuthClient and extend it to AAA_CreditControlClient (Figure B.1 (3)) to achieve the diameter client functionality described in section 2. In AAA_CreditControlClient, the TxCreditControlRequest function composes the `CCR` message and sends this message to the diameter server. The incoming answer messages from the diameter server are handled by the AnswerMsg virtual function. The function is overwritten in AAA_CreditControlClient to recognize the `CCA` message. Class DiameterAvpWidgetLibrary (Figure B.1 (7)) contains the Attribute-Value Pairs (AVP) widgets, i.e., the wrapper functions, which assist the composition and decomposition of the `CCR/CCA` messages. This class includes the credit control AVPs defined in RFC 4006, such as CC-Request-Type and Service-Context-Id AVP.

On the server side, we create an instance of AAA_CreditControlServer (Figure B.1 (4)), which is a sub-subclass of AAA_ServerAuthSession (Figure B.1 (6)). Similar to the diame-

ter client implementation, when the CCR message arrives, the overwritten virtual function RequestMsg handles this message. It reads the AVPs contained in this message and sends the response CCA message back to the client via the TxCreditControlAnswer function.

Note that the client and server session classes only provide diameter session management. The diameter message routing is contained in application class, which is not shown in Figure B.1 for simplification. By binding to the application class, the session class is able to send and receive messages from the routing capability provided by the application class. The detailed description of the application class can be found in [38].