

# Real-Time Billing Throughput Analysis of Wireless Telecommunication Systems

Jiacun Wang, *Senior Member, IEEE*

**Abstract**—Real-time billing is one of the most important functionalities in a telecommunication service system. In GPRS network, the load of billing traffic depends on the load of call traffic and the size of billing records. During busy hours, the GPRS network might not be able to transfer billing records on a timely basis if new billing records are generated too fast. This paper considers the capacity of the billing record collection and transfer system of the GPRS network. Stochastic timed Petri nets are employed to establish a template for the analysis of the capacity versus various factors.

**Index Terms**—Wireless telecommunication, GPRS network, throughput, billing, stochastic timed Petri nets

## I. INTRODUCTION

THE General Packet Radio Service (GPRS) is a new non-voice value added service that allows information to be sent and received across a mobile telephone network [2, 3]. It supplements today's Circuit Switched Data and Short Message Service. GPRS has several unique features. The most distinguishing one is its high speed. Its theoretical maximum speeds of up to 171.2 kilobits per second (kbps) is about three times as fast as the data transmission speeds possible over today's fixed telecommunications networks and ten times as fast as current Circuit Switched Data services on GSM networks. By allowing information to be transmitted more quickly, immediately and efficiently across the mobile network, GPRS provides a relatively less costly mobile data service compared to Short Message Services and Circuit Switched Data.

GPRS facilitates several new applications that have not previously been available over GSM networks due to the limitations in speed of Circuit Switched Data (9.6 kbps) and message length of the Short Message Service (160 characters). GPRS fully enables the Internet applications we are used to on our desktop from web browsing to chat over the mobile network. Other new applications for GPRS, include

file transfer and home automation- the ability to remotely access and control in-house appliances and machines.

Billing is one of the most important functionalities of GPRS services. A billing record, which contains charging information and is referred as Call Detail Record (CDR) in wireless telecommunication society, is generated when a call is taken down, and it is then transferred to the billing system chosen by the network operators through a charging gateway [6]. The load of billing traffic depends on the load of call traffic and the size of a billing record. During busy hours, the GPRS network might not be able to transfer billing records on a timely basis if the network's data transferring threshold is reached. Therefore, it is important to know the network's billing information processing capacity. In this paper, a stochastic timed Petri net model is established to facilitate the capacity analysis. The paper is organized as follows: Section 2 briefly introduces the GPRS billing information collection mechanism. Section 3 describes the modeling and capacity analysis of the GPRS billing information collection system. Section 4 gives a concluding remark.

## II. BILLING FLOW OF GPRS

### A. Charging Gateway Functionality (CGF)

As shown in Fig. 1, the Charging Gateway Functionality (CGF) provides a mechanism to transfer charging information from the Serving GPRS Support Nodes (SGSN) and Gateway GPRS Support Nodes (GGSN) to the network operator's chosen Billing Systems (BS). The Charging Gateway concept enables an operator to have just one logical interface between the GSNs and the BS.

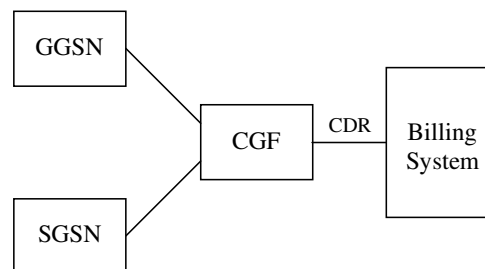


Fig. 1. Billing records collection flow.

Manuscript received September 30, 2004.

J. Wang is with the Monmouth University, West Long Branch, NJ 07764, USA. E-mail: jwang@monmouth.edu.

The CGF provides the mechanism to transfer charging information from the SGSN and GGSN nodes to the network operator's chosen Billing Systems(s) (BSs). The main functions of the CGF are:

- the collection of GPRS CDRs from the GPRS nodes generating CDRs;
- intermediate CDR storage buffering;
- the transfer of the CDR data to the billing systems.

In addition to the above main functionalities, the CGF may also perform some other activities, such as consolidation of CDRs, pre-processing of CDR fields, filtering of un-required CDR fields, and adding of Operator defined fields for specific billing systems. These specific activities may be performed to optimise the charging information that is to be forwarded to the Billing System, which should reduce the load in the Billing System.

The CGF can reside in a separate Network Element (Charging Gateway) or be integrated in the GSNs. It can receive CDR fields from the GSNs in real-time mode. It should have enough storage to enable it to transmit the collected charging data to the Billing System (BS) in file mode.

### B. CDR Generation

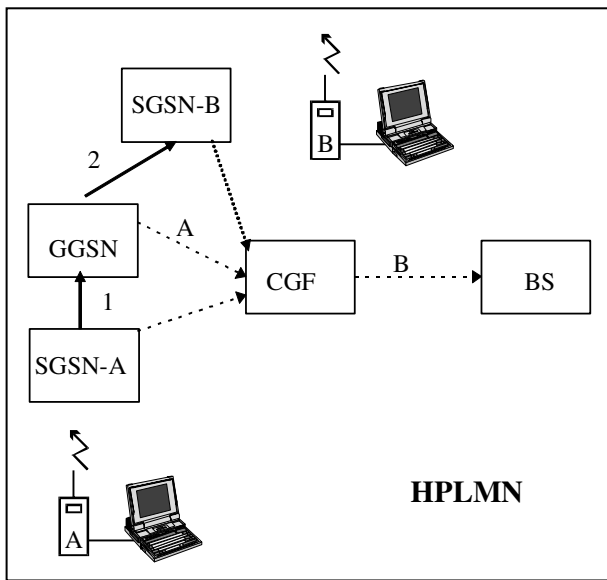


Fig. 2. CDR generation for GPRS mobile to mobile call [6].

Fig. 2 illustrates a simple GPRS mobile to mobile context within the same Home Public Land Mobility Network (HPLMN).

- The respective A-party related PDP context is activated in the SGSN-A and the GGSN (1).
- After the location of subscriber "B" is determined, the B party related PDP context is activated (2) in the SGSN-B and the GGSN and PDP PDUs are routed in MO and MT

direction. The SGSN-A shall create an S-CDR and the GGSN shall create a G-CDR for subscriber A, the SGSN-B shall create an S-CDR and the GGSN shall create a G-CDR for subscriber "B".

- If subscriber "A" and subscriber "B" use the same GGSN, both G-CDRs are produced at that GGSN.
- If session leg (2) requires a PDP context activation the respective PDP records will contain a network initiated PDP context activation-flag.
- The records generated are subsequently transferred to the CGF (A). The CGF transfers the CDRs to the BS.

### C. CDR Packet Transfer

Fig. 3 represents the default mode of CDR transfer from the CDR generating entities (GSNs) to the CDR packet collecting entities (CGFs).

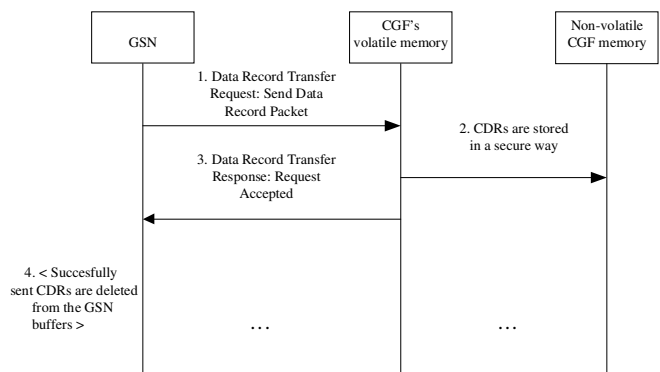


Fig. 3. CDR transfer from GSN to CGF.

- 1) The CDR generating entity (here the GSN symbolises either SGSN or GGSN) sends CDR(s) in a packet to CGF (that is the current primary Charging Gateway Functionality for the specific CDR generating node, "CGF1"). The sending is performed by using the Data Record Transfer Request message, with the Packet Transfer Command IE having the value 'Send Data Record Packet'.
- 2) The CGF opens the received message and stores the packet contents in a safe way (to e.g. a redundant RAM memory unit or a mirrored non-volatile memory or even to another node).
- 3) The CDR receiving entity (CGF) sends confirmation of the successful packet reception to the CDR generating node (GSN). The confirmation is performed by using the Data Record Transfer Response message, with the Cause value being 'Request Accepted'.
- 4) After the positive response 'Request Accepted' is received by the GSN, it may delete the successfully sent CDRs from its send buffer.

If the positive response has not been received within a configurable time-out limit, the GSN will retransmit the

request. The maximum amount of retries is a configurable value based on the GSPN transport protocol.

#### D. CGF Downstream File Transfer

The CGF downstream file transfer application transfers billing files from the CGF to a downstream system. It can transfer all available billing files in chronological order, including the currently open billing file, using FTP protocol.

#### E. Backup Mode of CDR Transfer

CGF billing throughput capacity includes many factors. The principal indicators are the *Busy Hour Call Attempts* (BHCA), and the billing record size. For the CGF this translates in how much billing data throughput can be processed (in Bytes/s). Other factors include the need to reserve CGF resources to recover from fault conditions.

Whenever GSN billing data cannot be transferred to the CGF, CGF goes into a *backup* mode - the billing data is written in temporary files to disks on the GSN. The data is not stored redundantly in this mode. Once GSN is able to send billing data to the CGF, GSN goes into a recovery mode - in addition to sending the current billing records from the GSN to the CGF, the billing data from the backup files is also sent.

#### F. CGF Real-Time Billing

With streams configured to store the received billing records on the CGF, and configured for scheduled outbound file transfer, CGF supports a capability referred to as Real-Time Billing. This capability will transfer data downstream from the currently open billing file, as it is being written to the CGF disk, containing the most current billing records received from the GSN. The downstream system can thereby process the billing records received by the CGF in near real-time.

Note that following a backup incident on the GSN, the records transferred from the GSN as part of the recovery sub-stream are not delivered by Real-Time Billing; they will be transferred at the scheduled interval. The scheduled file transfer will also transfer any other closed files that were not sent by the real-time billing transfer process; this can occur, for example, when there is a network outage to the downstream system.

### III. BILLING THROUGHPUT ANALYSIS

Billing capacity is one of a wireless service provider's most concerns. In this section we use stochastic timed Petri nets to model and analyze the billing capacity.

#### A. Stochastic Timed Petri Nets

A Petri net is a graphical and mathematical modeling tool [7]. It consists of *places*, *transitions*, and *arcs* that connect them. *Input arcs* connect places with transitions, while *output arcs* start at a transition and end at a place. Places can contain *tokens*; the current state of the modeled system (the *marking*) is given by the number (and type if the tokens are

distinguishable) of tokens in each place. Transitions are active components. They model activities which can occur (the transition *fires*), thus changing the state of the system (the marking of the Petri net). Transitions are only allowed to fire if they are *enabled*, which means that all the preconditions for the activity must be fulfilled (there are enough tokens available in the input places). When the transition fires, it removes tokens from its input places and adds some at all of its output places. The number of tokens removed / added depends on the cardinality of each arc.

Petri nets are a promising tool for describing and studying systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. As a graphical tool, Petri nets can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. In addition, tokens are used in these nets to simulate the dynamic and concurrent activities of systems. As a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behavior of systems. The characteristics exhibited by the activities of information processing systems such as concurrency, decision making, synchronization and priorities are modeled very effectively with Petri nets.

To study performance and dependability issues of systems it is necessary to include a timing concept into the model. There are several possibilities to do this for a Petri net; however, the most common way is to associate a *firing delay* with each transition. This delay specifies the time that the transition has to be *enabled*, before it can actually fire. If the delay is a random distribution function, the resulting net class is called *stochastic timed Petri nets* (STPNs) [1, 5, 9]. Different types of transitions can be distinguished depending on their associated delay, for instance *immediate transitions* (no delay), *exponential transitions* (delay is an exponential distribution), and *deterministic transitions* (delay is fixed).

STPNs as a formalism developed in the field of computer science have gained wide applications for modeling system performance. A lot of software tools have been written to implement and solve models defined using STPN formalism.

#### B. STPN Modeling of GPRS Billing Flow in GSN Side

Fig. 4 shows the STPN model of CDR generation, store and transfer in the GSN side. All nodes are explained in Table 1. The model is composed of 8 transitions and 9 places. Place *WSIZE* models the sliding window size for the control of CDR packets transfer from GSN to CGF, and its capacity and initial marking are set to  $n$ , where  $n$  is the window size. Place *BUF* models the buffer threshold for switching the application from normal mode to backup mode; its capacity is set to  $m$ , while its initial marking is 0. The weight of directed arcs (*BUF*, *bklg*) and (*bklg*, *BUF*) are all set to  $m$ , meaning when there are  $m$  tokens accumulated in place *BUF*, transition *bklg* will fire. The firing of *bklg* will remove  $m$  tokens from and then put them back to *BUF* and put one token to place *BKLG*, meaning the system is in backlogged mode.

In normal mode (place *NORM* is marked), when call processing entity generates a billing record (transition *call* fires, and a token is added to place *CINS*), a process in the GSN will create a CDR packet and put it to the packet buffer pool (transition *gCdr* fires, a token added to each of places *BUF* and *PACK*). Then a second process sends the record to the CGF side through network link (transition *send* fires, a token added to *TOCGF*). On the CGF side, when the data from GSN is processed successfully, an ACK is sent to the GSN (a token is added to place *ACK*), and the GSN will remove the original data from buffer pool upon receiving the ACK (*rmCdr* fires, a token added to *WSIZE*). Notice that before an ACK is received by the GSN, it can only send up to  $n$  packets to the CGF due to the existence of place *WSIZE* of capacity  $n$ .

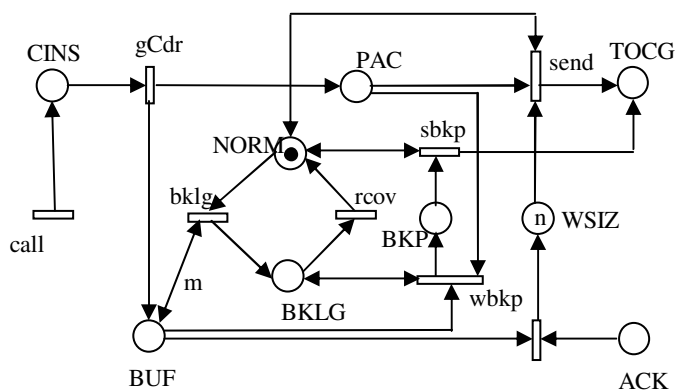


Fig. 4. STPN model of CDR generation, store and transfer in the GSN side

Table 1 Legend for Fig. 4.

Place	Description	Capacity
CINS	New call instance	
BUF	CDR packet in buffer	
PACK	CDR packet to transmit	
NORM	Normal transfer mode	
BKLK	Backlogged mode	
BKP	Backlogged packets	
WSIZE	Window size of packet transfer	$n$
TOCGF	Packet on the way to CGF	
ACK	ACK from CGF to GSN	
Transition	Description	
Call	New call	
gCdr	Generate new CDR packet	
bklg	Change to backlog mode	
rcov	Recovers from backlogged mode	
send	Send normal packet	
sbkp	Send backlogged packet	
wbpk	Write packets to backup files	
rmCdr	GSN removes ACK'ed records from bufpool	

If the CGF cannot catch up with the input from the GSN, more and more records will be accumulated in the buffer pool. If the number of accumulated records reaches the preset threshold (place *BUF* has  $m$  tokens or more), the CGF switches to backup mode (*bklg* fires, a token added to *BKLK*). The firing of *bklg* consumes the token in place *NORM*, which

disables transition *send*, indicating the normal packet transfer is halted. But the firing of *bklg* does not consume any token from *BUF*, as the switch of mode does not change the number of billing records in the buffer. After the firing of *bklg*, transition *wbpk* is enabled, indicating the system starts to write CDRs to a local disk. In this mode, no record will be transferred to CGF, because neither transition *send* nor transition *sbkp* is enabled. After certain time recovery mode occurs (transition *rcov* fires, and place *NORM* marked again), the data in backup file will join the main billing stream to be transferred to the CGF, and both of transitions *send* and *sbkp* are friable now.

### C. STPN Modeling of Billing Flow in CGF Side

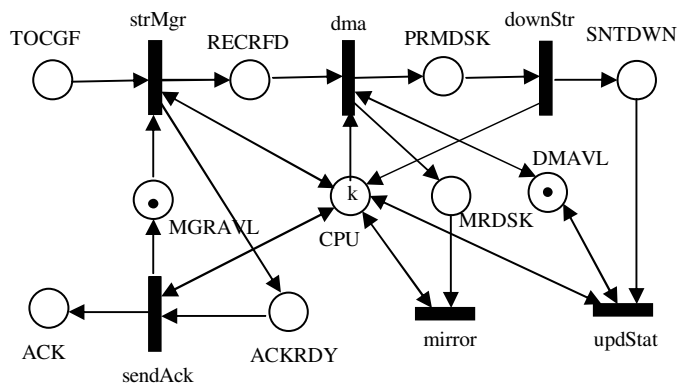


Fig. 5. STPN model of CDR transfer and processing in the CGF side.

Table 2 Legend for Fig. 5.

Place	Description	Capacity
TOCBM	Billing records in transit to CBM	
RECFRD	Billing records reformatted	
PRMDSK	Billing records in CGF primary disk	
MRDSK	Waiting to write billing records to mirrored disk	
SNTDWN	Records sent to downstream	
DMAVL	DMA process available	1
ACKRDY	Ready to send ACK	
CPU	CPU available	$k$
MGRAVL	Stream Manager available	1
ACK	ACK to GSN	
Transition	Description	
strMgr	CBM stream manager processes records	
dma	DMA writes records to disk	
dwnStr	Deliver data to downstream	
updStat	Update records attribute to 'sent'	
mirror	DMA writes records to mirrored disk	
sendAck	Send ACK	

Fig. 5 shows the STPN model of CDR transfer in the CGF side. Nodes are described in Table 2. The model is composed of 6 transitions and 10 places. (Notice that places *TOCGF* and *ACK* are duplicated here for the purpose of showing the connection between the GSN model and the CGF mode.) All transitions are timed transitions. Place *CPU*'s capacity and initial marking are  $k$ , meaning there are  $k$  CPUs in the CGF.

Places *MGRAVL* and *DMAVL* represent *streamManager* process, which is responsible for retrieving and reformatting CDRs from the buffer, and *DMA* process, which stands for direct memory access and is responsible for writing records to storages, respectively; their capacities are all one.

When a batch of billing record data is sent to CGF (a token added to *TOCBM*), on the CGF side, the stream manager picks up the record data and reformats it (*strMgr* fires, a token added to *RECRFD* and *ACKRDY* each). Then the *DMAPROC* process writes the billing data to the CGF primary disk (*dma* fires, a token added to *PRMDSK*). Now, downstream delivery process will transfer the data to downstream processor (*downStr* fires, a token added to *SNTDWN*), and the record data will be marked “sent” (*updStat* fires). Meanwhile, the same data will be written to the mirror memory, i.e. non-volatile memory (mirror fires). When the data from GSN is processed by stream manager, an ACK is sent to the GSN (*sendAck* fires, and a token is added to *ACK*).

#### D. Billing Throughput

We want to know the maximum CDR throughput of the billing flow. Notice that in the GSN side STPN model depicted in Figure 4, if the number of tokens in place *BUF* reaches its capacity, a backup mode occurs, in other words, the billing flow reaches the network’s maximum throughput.

We can find the billing throughput either based on theoretical analysis [8], or through simulation. Here we provide the simulation results by using the tool STPNPlay [10]. Assume the firing rates of time transitions are as follows:

$$\begin{aligned} r(\text{send}) &= r(\text{strMgr}) = r(\text{dma}) = r(\text{downStr}) = 0.002, \\ r(\text{bklg}) &= r(\text{sbkp}) = r(\text{wbkp}) = r(\text{rmCdr}) = 0.0005, \\ r(\text{sendAck}) &= r(\text{mirror}) = r(\text{updStat}) = 0.0005, \\ r(\text{rcov}) &= 5, \\ r(\text{gCdr}) &= r(\text{call}). \end{aligned}$$

Also assume  $k = 2$ ,  $m = 50$  and  $n = 20$ . The simulation covers 10000 steps of transition firings. Fig. 6 shows the number of tokens vs. transition firings when  $r(\text{call}) = 0.0035$ . At this billing rate the billing flow works in a normal mode, i.e., no backup mode happens, because the token number is always below 50. But if we increase the billing rate, backup mode will occur immediately. Fig. 7 shows the situation when  $r(\text{call}) = 0.0033$ , that is, the backup occurs periodically. So in this example, the maximum throughput of the billing flow is  $3600/0.0033 = 1.02$  million CDRs per hour.

#### E. Discussion

The STPN model can be improved by taking into account a couple of practical factors of the GPRS billing flow.

The first one we need to consider is the granularity of data blocks transferred from the GSN to the CGF. Let us say according to the communication protocol between the GSN and the CGF, each time the GSN can transfer up to  $b$  bytes of data to the CGF, while the CDR size is  $c$  bytes. Therefore, the GSN will transfer  $\text{int}(b/c)$  CDRs each time to the CGF, instead of one CDR. To incorporate this factor into our

model, we only need to change the weight of the arcs ( $\text{gCdr}$ ,  $\text{PACK}$ ) and ( $\text{gCdr}$ ,  $\text{BUF}$ ) to  $\text{int}(b/c)$ .

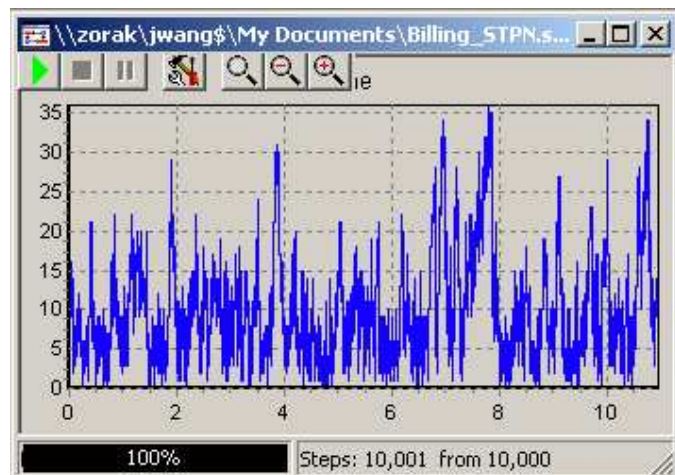


Fig. 6. Number of tokens in place *BUF* when  $r(\text{call}) = 0.0035$

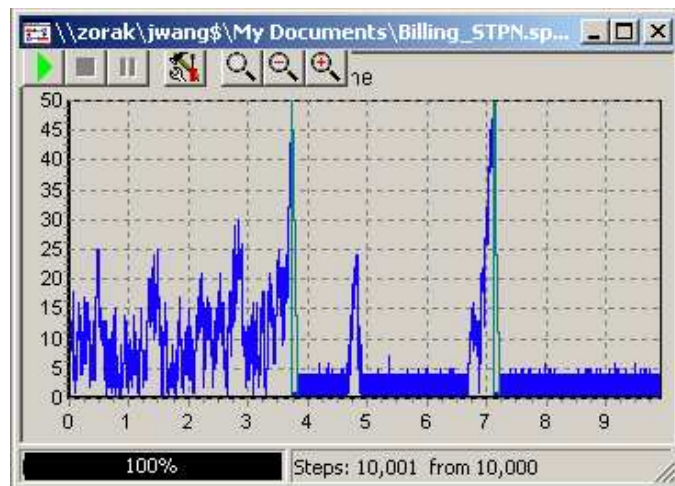


Fig. 7. Number of tokens in place *BUF* when  $r(\text{call}) = 0.0033$ .

Another one may have impact to the billing throughput is the sizes of CDRs generated from GGSN and SGSN. If the sizes of these two types of CDRs are different, then the processing and transfer times of these two types of CDRs will be different. In this case, we have two options to improve the model. The first one is to get the average CDR size assuming that the percentages of these two types of CRDs in the traffic are known. Then we can decide the processing and transfer times based on the average CRD size, and there is no need to change the models we built. The second approach is to deploy the colored Petri net modeling skills and differentiate types of CRDs using the colors of tokens [4]. The transitions firing times will also be different when they are triggered by different types of tokens.

#### IV. CONCLUDING REMARKS

This paper presented a modeling and analysis approach to the real-time billing flow in a GPRS wireless telecommunication service based on the STPN formalism. In a GPRS, the load of billing traffic depends on the load of call traffic and the size of a billing record. During busy hours, the GPRS network might not be able to transfer billing records on a timely basis if new billing records are generated too fast and the network's data transferring threshold is reached. Therefore, it is important to know the network's billing information processing capacity.

Although this paper focuses on GPRS billing flow analysis, the same approach can be applied to GSM and UMTS billing flow analysis as well.

#### REFERENCES

- [1] M. Ajmone Marsan, G. Balbo, G. Conte, A class of generalized stochastic Petri nets for the performance analysis of multiprocessor systems, *ACM Trans. on Computer Systems*, 2(1), 93-122, May 1984.
- [2] G. Heine and H. Sagkob, *GPRS: Gateway to Third Generation Mobile Networks*, Artech House, Incorporated, 2003
- [3] A. Kavanagh and J. Beckmeyer, *GPRS Networks*, Osborne Publishing, 2002
- [4] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, Springer-Verlag Telos, 1996.
- [5] M. Molloy, Performance analysis using stochastic Petri nets, *IEEE Transactions on Computers*, C-31(9), 913-917, 1982.
- [6] 3GPP Specifications, TS 32.015, <http://www.3gpp.org/ftp/Specs/html-info/32-series.htm>
- [7] J. Wang, *Timed Petri Nets: Theory and Application*. Boston: Kluwer Academic Publishers, 1998.
- [8] J. Wang, M. Zhou and Y. Deng, Throughput analysis of discrete event systems based on stochastic Petri nets, *International Journal of Intelligent Control and Systems*, vol. 3, no. 3, 343-358, 1999.
- [9] W.M. Zuberek, Timed Petri nets: definitions, properties, and applications, *Microelectron. Reliab.*, 31(4) 627-644, 1991.
- [10] <http://dce.felk.cvut.cz/capekj/StpnPlay/>

**Jiacun Wang** (SM'2000) received his Ph.D degree in Computer Science for Nanjing University of Science and Technology in 1991.

He is currently an Associate Professor with Department of Software Engineering, Monmouth University, New Jersey, USA. He previously worked for Nortel Networks as a Scientific Staff Member in the Wireless Telecommunication Department. His research interests include software architecture, formal methods, real-time systems and wireless networking.