Performance Analysis of a Mobile Network for API-Oriented Traffic

Daizo Ikeda[1], Toshihiro Suzuki[1] and Akira Miura[2]

[1]*Research Laboratories, NTT DOCOMO, Inc.*
3-5 Hikarino-oka, Yokosuka-shi, Kanagawa 239-8536, Japan
[2]*Prefectural University of Kumamoto*
3-1-100, Tsukide, Kumamoto, 862-8502, Japan
{ikeda, suzukitoshi}@nttdocomo.co.jp, miura@pu-kumamoto.ac.jp

Abstract

In aiming for service innovation, mobile operators are in the process of providing an environment that encourages third parties to create a wide variety of Web applications by making use of an application programming interface that the network offers. This paper presents an analysis method for dealing with API-oriented traffic, one of the major operational challenges in the coming IMS era which must be overcome in developing a highly stable and reliable communication system. We addressed this issue by extending traffic evaluation methods used for the commercial i-mode service in Japan. API traffic models are defined based on process sequences and the user packets invoked by an API request and these models are used in the performance evaluation of nodes in a system. We found that API traffic can have a large impact on the CPU use of the gateway modules. Thus, network capacity planning should take account of the impact of estimated API traffic on a mobile network. Our proposal allows mobile operators to construct a highly stable and reliable system which supports service innovation by providing APIs to application developers.

Keywords: *mobile communications; traffic patterns; performance evaluation; API; IMS*

1. Introduction

Over the past decade, mobile networks have been developed widely, in many countries, to provide an environment for the mobile Internet and today are moving into the next stage of service innovation. Service enablers on a mobile platform will encourage the creation of a wide variety of services and applications by opening up an application programming interface (API) to the Internet. Open APIs enable application developers to make use of functions useful for service development, such as short messaging, and to acquire information, including terminal location and presence. Some mobile operators, such as Vodafone and Telefonica, are offering network APIs to their collaborative developers and so seeking innovative service development [1, 2]. Furthermore, many researches also have been conducted in this field. While more operators, researchers and developers have expressed interest in these opportunities for innovation, some activities to promote standard-setting for APIs have also emerged. One of the major standard bodies is Open Mobile Alliance (OMA), which defines Parlay X and Next Generation Service Interface (NGSI) [3, 4]. However, the introduction of the API requests that result from these new services makes it difficult to estimate the amount of traffic in a mobile network, even though many studies have been conducted in the traffic management domain [5, 6]. To address this, we have developed a technique to extend the traffic analysis method used for a legacy second generation mobile

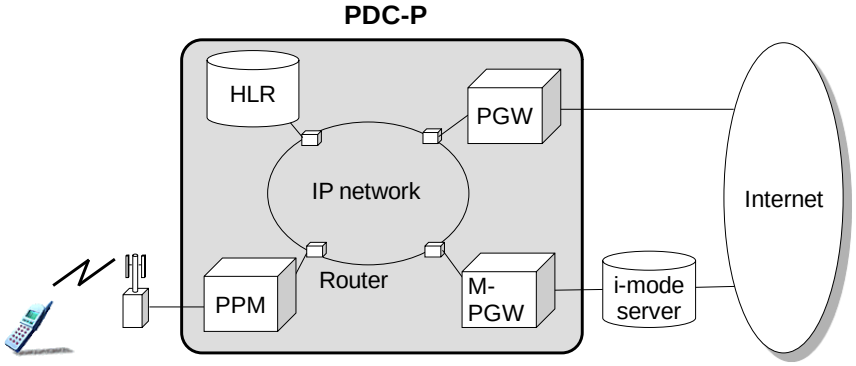
network supporting the i-mode service to an API-equipped mobile network for the coming IMS era.

In this paper, we presents a concept for developing a high quality communication system for IMS/Web 2.0 based on developing techniques used for an i-mode core network. This paper is organized as follows. Section 2 describes major operational challenges in the legacy second generation mobile network. Section 3 describes practical approaches to overcoming these challenges. Section 4 proposes a method for API-oriented traffic analysis and Section 5 examines the results of performance evaluations and consequent implications for network capacity planning. Section 6 provides concludes relating to our proposal.

2. Operational Challenges in the PDC-P

The i-mode service was launched in 1999 by NTT DOCOMO, INC., a leading mobile operator in Japan, and currently has more than fifty million subscribers. The aim of this service is to be at the forefront of the mobile Internet by creating an environment which provides easy-to-use e-mail operation and Internet access. The i-mode service was widely accepted as an innovative tool for getting access to the Internet and acquired more subscribers than expected within a relatively short time period, and thus was confronted with operational challenges due to overload.

The i-mode service is provided over a large-scale network of the second-generation mobile communication system called personal digital cellular-packet (PDC-P) [7, 8]. As shown in Figure 1, this system mainly comprises a large number of PDC-P nodes, namely, packet processing modules (PPMs), packet gateway modules (PGWs), and mobile message packet gateway modules (M-PGWs) [9]. PGWs are connected to the Internet, as are M-PGWs via i-mode servers. All of the network nodes are Unix-based and transfer data packets over the IP network. Multiple M-PGWs are installed to balance the traffic load and improve reliability. The i-mode service allows information transfer between a mobile station and a content server on the Internet over HTTP [10].



PPM: Packet processing module
PGW: Packet gateway module
M-PGW: Mobile message PGW
HLR: Home location register

Figure 1. PDC-P Network

For the purpose of creating an environment for mobile Internet, the highest priority was placed on an early launch of i-mode and for this reason scalability was planned to be achieved

at a later stage. However, beyond market expectations, the number of mobile subscribers rapidly grew to one million, and then to ten million within one year from commencement of service. At the peak period, the number of newly-acquired subscribers reached more than fifty thousand per day. As the amount of traffic increased to the point of overload, hardware and software bottlenecks emerged in the PDC-P network resulting in occasional service interruptions due to software problems. Immediate measures were taken to not only bolster the hardware, but also to improve the software in terms of robustness and maintenance of quality in overcoming the overloads. Furthermore, an elaborate survey was conducted to estimate the number of network elements required, in order to determine the need for future upgrades to equipment such as to servers, routers, and switching units in a timely and effective manner. The following sections describe practical strategies that we adopted to meet such needs.

3. Practical Approaches to Overcoming Three Challenges

3.1. Identification of Traffic Characteristics

It is important to note that there are two major characteristics of mobile data communications. One is that users move around and the other is that the throughput tends to fluctuate because of the existence of wireless sections. Another factor to consider is the characteristics of packet switching: data packets occur at random while a terminal is in a communication mode. We found that the packet call distribution is similar to voice call distribution based on measurements of the node traffic in the PDC-P network and the traffic between M-PGWs and i-mode servers [11]. In general, the precision of a distribution is evaluated using a decision coefficient. This was more than 0.9 in the actual measurement, which indicates strongly that the i-mode traffic pattern follows an exponential distribution. Figure 2 shows the measurement results for the number of requests to initiate i-mode communication as a result of a packet arriving in the network for delivery to a mobile terminal (hereafter called “packet arrival”).

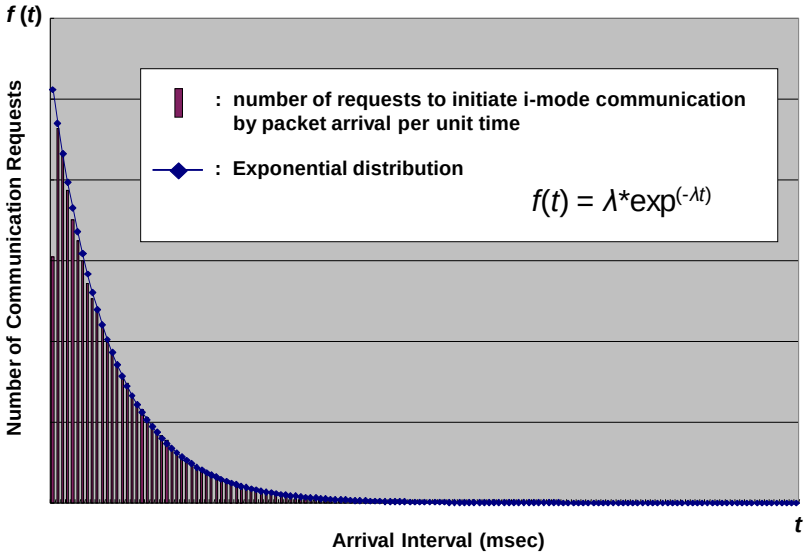


Figure 2. i-mode Traffic Pattern

3.2. Identification of Bottlenecks

One of the major targets was to enhance the capacity in order to eliminate hardware and software bottlenecks, which are primarily due to the sharply increasing amount of traffic. To avoid processing bottlenecks which may result from simultaneous attempts to evaluate the performance of and to upgrade software, a physically independent performance evaluation site and overload simulators were newly constructed in an already-constructed debugging environment as shown in Figure 3. Furthermore, through fixed point observation of traffic patterns, and the gathering and analysis of processing logs on the commercial PDC-P units, we identified a number of bottleneck points. This allowed us to perform recovery and effectiveness tests. As a result of the testing, we successfully reduced the number of software update failures of the commercial i-mode system, which might result from errors in signals or process sequences. Thus, by evaluating the effectiveness of software upgrades before introducing them into the commercial system, we successfully improved the robustness against overloads.

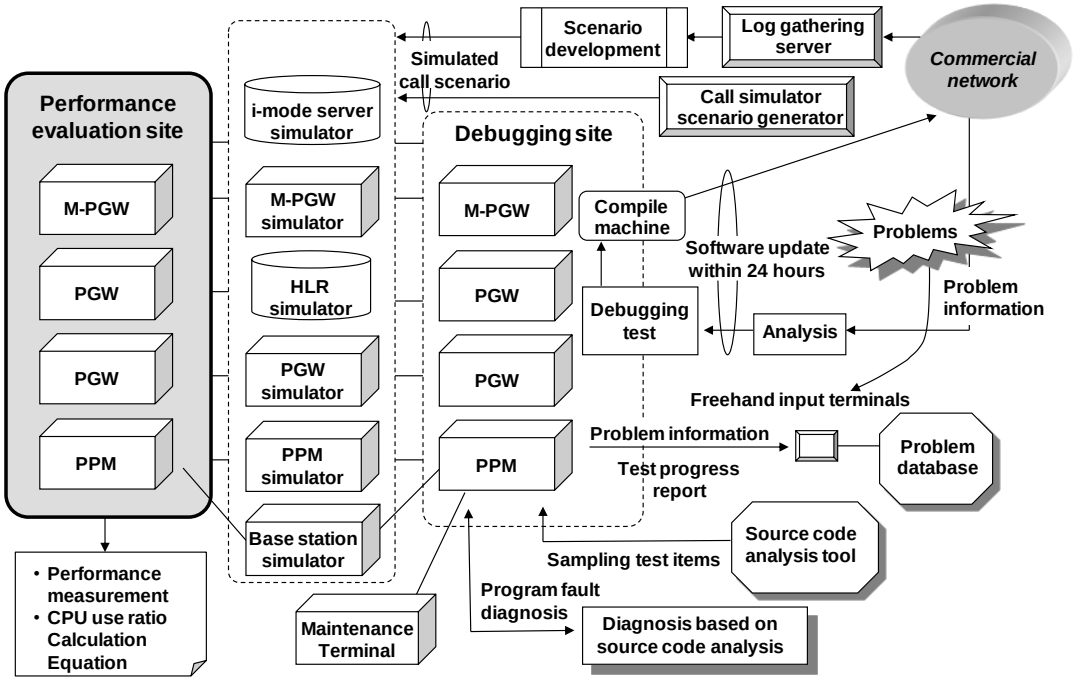


Figure 3. Development Environment

3.3. Performance Evaluation

We recorded measurements under overload and rapidly-changing traffic conditions at the performance evaluation site, and conducted a regression analysis of the CPU use ratio for the servers. The performance evaluation equation is defined as

$$CPU_{Occupancy} = \sum_{i=1}^{n} \alpha_i X_i \quad (1)$$

where α_i stands for coefficients representing the weight for each process, such as, user packet transfer, communication requests, and channel switching and X_i stands for parameters indicating the number of times the process is performed [12]. We selected the signals that affect the processing capacity, including call control signals and maintenance signals [13], and extracted the n parameters that indicate the number of times that a certain process is executed based on the number of signals processed. The coefficients were calculated in a single regression analysis by generating a traffic load specific to a particular coefficient at the performance evaluation site. For example, the coefficient for user packet transfer is calculated by generating a user packet transfer load exclusively. Similarly the coefficient for each other type of process is calculated by applying the load for that particular process. Thus, a number of measurements are conducted to cover all parameters.

This technique clarified the performance limits and defined highly accurate criteria for increasing or decreasing the number of network elements [14]. One benefit was an improvement in the overload robustness. We clearly detailed procedures for replacing PDC-P nodes and switching units with higher capacity units by optimizing the thresholds and the timing of the upgrades. Also, the performance evaluation results were applied to optimize the congestion control.

4. Proposed Method for API-Oriented Traffic Analysis

In the IMS network, a mobile operator is expected to provide service enablers to application developers through open APIs. Application developers may be allowed to create Web applications using network resources and functions by using APIs. As a result of this collaborative innovation process, a wide variety of traffic is expected to pour into the communication system, including requests for terminal location information or normal call initiation, from Web applications using an API. In such an environment, the observation of traffic and the identification of its characteristics will become increasingly important. Different services generate different traffic characteristics. However, one essential approach will be to investigate the traffic patterns of the major types of traffic and to evaluate their impact on the mobile network. The following is our analysis framework for estimating the impact of API-oriented traffic on a mobile network.

(1) API-oriented traffic models were defined based on process sequences.

(2) API-oriented traffic volumes were defined based on conventional commercial traffic.

(3) Under the models, the CPU use ratios were calculated by using a performance evaluation formula.

(4) The evaluation results were analyzed for implications on network capacity planning.

The first step is to investigate the traffic patterns of major service types. Some standards bodies, including OMA, define specifications for providing APIs. One of these is Parlay X web services, which are also defined as 3rd Generation Partnership Project (3GPP) standards, as shown in Table 1 [15]. Parlay X web services are designed for Web applications and free developers from the need to understand complicated communication protocols by implementing protocol conversion to a Parlay API. Parlay X works in a Web Services environment, which allows application developers to use it in a web-like manner over the Internet. Taking into account the API specifications, we defined three models based on how a core network may process an API request, as shown in Table 2.

Table 1. Specifications of Parlay X Web Services

3GPP Specification series	Open Service Access (OSA); Parlay X web services
TS 29.199-01	Part 1: Common
TS 29.199-02	Part 2: Third party call
TS 29.199-03	Part 3: Call notification
TS 29.199-04	Part 4: Short messaging
TS 29.199-05	Part 5: Multimedia messaging
TS 29.199-06	Part 6: Payment
TS 29.199-07	Part 7: Account management
TS 29.199-08	Part 8: Terminal status
TS 29.199-09	Part 9: Terminal location
TS 29.199-10	Part 10: Call handling
TS 29.199-11	Part 11: Audio call
TS 29.199-12	Part 12:Multimedia conference
TS 29.199-13	Part 13: Address list management
TS 29.199-14	Part 14: Presence
TS 29.199-15	Part 15: Message broadcast
TS 29.199-16	Part 16: Geocoding
TS 29.199-17	Part 17: Application-driven Quality of Service (QoS)
TS 29.199-18	Part 18: Device capabilities and configuration
TS 29.199-19	Part 19: Multimedia streaming control
TS 29.199-20	Part 20: Multimedia multicast session management
TS 29.199-21	Part 21: Content management
TS 29.199-22	Part 22: Policy

Table 2. API-oriented Traffic Models

	Definition	Examples
Model 1	**Communication initiation triggered by an API request**	**Third party call, short messaging, multimedia messaging**
Model 2	**Server access to acquire information triggered by an API request**	**Terminal status, address list management**
Model 3	**User packet transferred incorporating an API request**	**Terminal location, device capabilities and configuration**

Here, Model 1 defines the case where an API request triggers communication initiation. We assumed that the number of call control signals to initiate communication triggered by an API request is as shown in Figure 4. Examples include Third Party Call API and Short Messaging API. Model 2 defines the case where an API request triggers an access to subscriber-related or terminal-related information held on a database in the core network. We assumed the number of call control signals to acquire the requested information triggered by an API request is as shown in Figure 5. These include Terminal Status API and Address List

Management API. Model 3 is designed to evaluate the impact of user packets conveying an API request and so cannot be readily identified. An example is Terminal Location API.

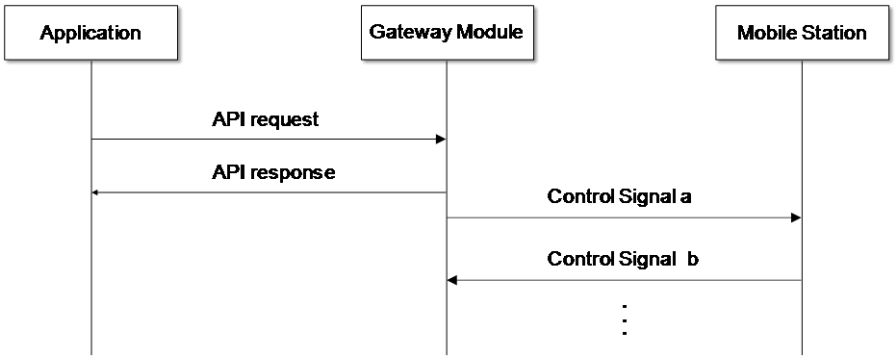


Figure 4. Signals Triggered by an API Request – Model 1

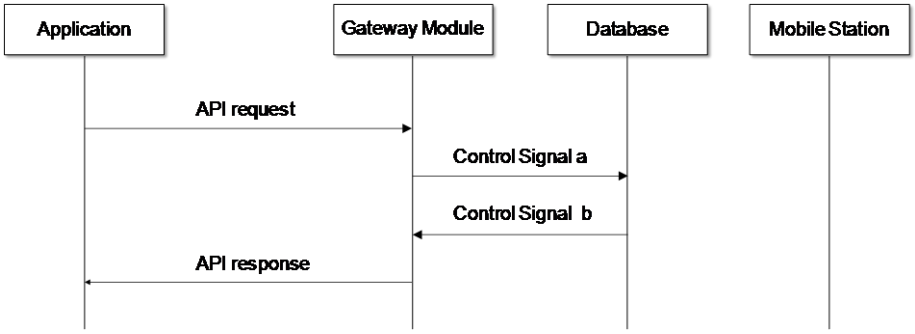


Figure 5. Signals Triggered by an API Request – Model 2

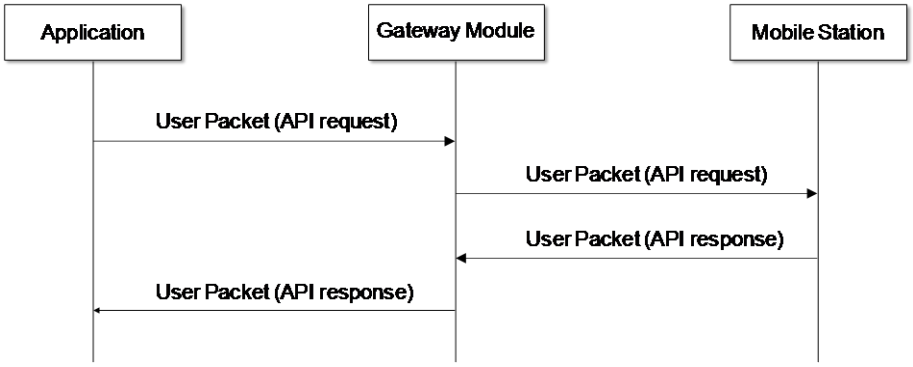


Figure 6. Signals Triggered by an API Request – Model 3

The second step is to assume an API-oriented traffic volume based on conventional commercial traffic. We assumed that the API request distribution is similar to the packet call distribution, which follows an exponential distribution, as shown in Figure 2. This is because an API request in Model 1 initiates communication. Even though an API request in Model 2 does not itself trigger communication initiation, the acquired information such as terminal

status may be partly used to initiate communication afterwards. Furthermore, we defined certain cases of traffic volumes to evaluate the impact of increasing API requests on the mobile network. Non-API traffic models were also defined for comparison with the API-oriented cases. Section 5 details how we determined certain traffic volumes for each model.

The next step is to evaluate the effects on the gateway modules in the mobile network by distinguishing arriving call packets from service requests from an API. Performance analysis techniques originally used for the i-mode traffic were applied to the API-oriented case. The performance evaluation equations corresponding to Model 1 mentioned above were defined as

$$CPU_{MODEL1} = \alpha_1 userpacket + \alpha_2 initiation\_byPA + \alpha_3 initiation\_byMS + \alpha_4 initiation\_byAPI + \alpha_5 dbaccess\_byAPI + \alpha_6 userpacket\_byAPI \quad (2)$$

where α_i stands for coefficient representing the weight applied to process i and the various parameters represented the number of incidents of a specific process as follows.

userpacket: user packet conveying content

initiation_byPA: communication initiation triggered by packet arrival

initiation_byMS: communication initiation triggered by a mobile station.

initiation_byAPI: communication initiation triggered by an API request

dbaccess_byAPI: information access triggered by an API request

userpacke_byAPI: user packet conveying an API request

The coefficients for conventional traffic were applied to the API-oriented cases. The performance evaluation formulae for Model 2 and Model 3 were defined in the same way. The differences are that Model 2 evaluates the impact of traffic for information access triggered by an API request and Model 3 addresses user packet conveying an API request.

$$CPU_{MODEL2} = \alpha_1 userpacket + \alpha_2 initiation\_byPA + \alpha_3 initiation\_byMS + \alpha_5 dbaccess\_byAPI \quad (3)$$

$$CPU_{MODEL3} = \alpha_1 userpacket + \alpha_2 initiation\_byPA + \alpha_3 initiation\_byMS + \alpha_6 userpacket\_byAPI \quad (4)$$

5. Evaluation Results

5.1. Performance evaluation of Model 1

For the purpose of evaluating the impact of Model 1 traffic on the core network, API traffic models were defined based on the traffic patterns for communication requests due to packet arrival. Here, three API traffic models were used to evaluate the performance when subject to the estimated API requests, along with the two non-API models, and these were defined based on the conventional commercial traffic of call requests. Medium API traffic means that the volume of API requests is equivalent to the number of communication requests due to packet arrival per second. Two non-API models, which offer no open APIs, were used to evaluate the performance of two cases, namely, user packet transmission alone, and both

user packets and call control signals. The volumes of user packets and call control signals were determined based on commercial traffic models. In addition,

Consequently, the CPU use ratio was calculated by applying a performance evaluation formula (2) as described in section 4. The results, shown in Figure 7, made it clear that API-oriented traffic models led to high CPU occupancy, reaching the threshold of 60%, which is commonly accepted as a limit for stable operation, with a smaller number of user data packets transferred than was the case with non-API traffic models [16].

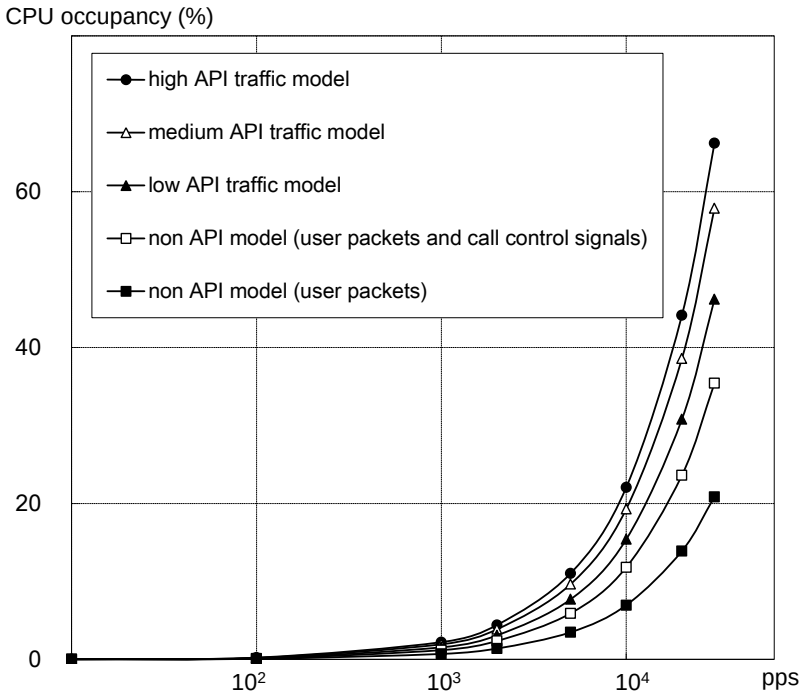


Figure 7. Performance Evaluation Results for Model 1

This result implies that API-oriented traffic should be reflected in network capacity planning, along with traffic for the normal initiation of communication. One example is the planning of gateway modules, which process both the initiation of communications triggered by packet arrival and API requests. Based on assumptions of API request volumes, it is critical to estimate the possible transferred user packet data for stable CPU use. Such performance evaluations should also make it possible to clarify the need for exclusive gateway modules to process API requests.

5.2. Performance Evaluation of Model 2

Using Model 2, we evaluated the performance of the estimated API requests for access to information databases in the core network, including terminal status and address list databases. Three API traffic models and two non-API models were used. For this model, medium API traffic means that the volume of API requests is equivalent to the number of communication requests due to packet arrival per second, while high API traffic is eight times greater than this. Two non-API models, which assume no open APIs, were used for comparison with the two API models, and these were defined in the same way as for Model 1.

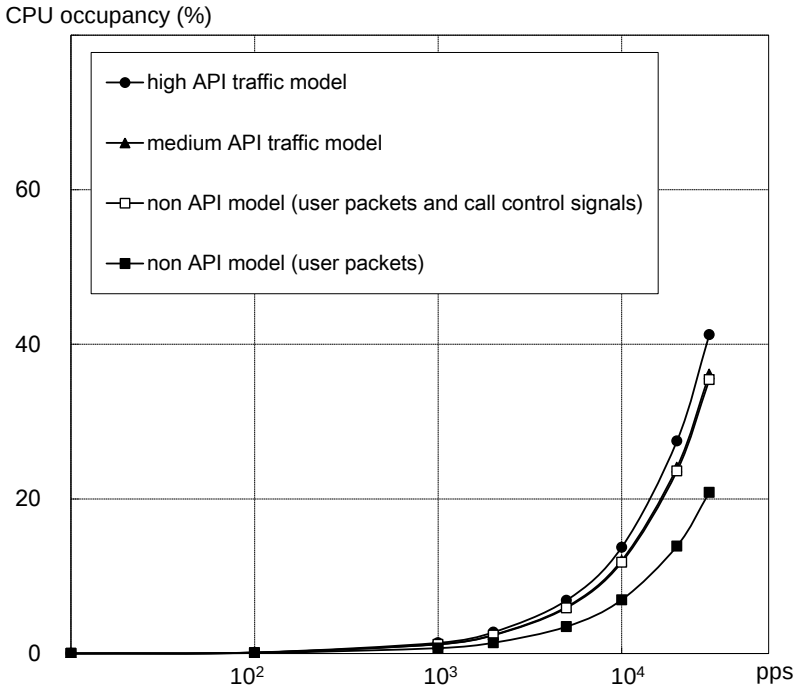


Figure 8. Performance Evaluation Results for Model 2

The CPU use ratio was calculated by applying a performance evaluation formula (3) as described in section 4. The results, shown in Figure 8, imply that API-oriented traffic, along with traffic for the information access of database in the core network, would have a slight impact on the network performance. We concluded that the estimated impact of high API traffic for information access should be reflected in network capacity planning.

5.3. Performance Evaluation of Model 3

Using Model 3, we evaluated the performance with regard to the estimated user packets conveying an API request and hence cannot be so easily identified. API requests for terminal location or device capabilities and configurations are sent over HTTP and thus it is important to evaluate the effects of user packets conveying an API request on the core network. We defined four traffic models with a different percentage of user packets conveying an API request. Medium API traffic means the ratio of API request packets to normal user packets is one to one, while high API traffic means a ratio of eight to one. The all API model means that all the user packets transferred in the core network convey an API request. The CPU use ratio was calculated by applying a performance evaluation formula (4) as described in section 4.

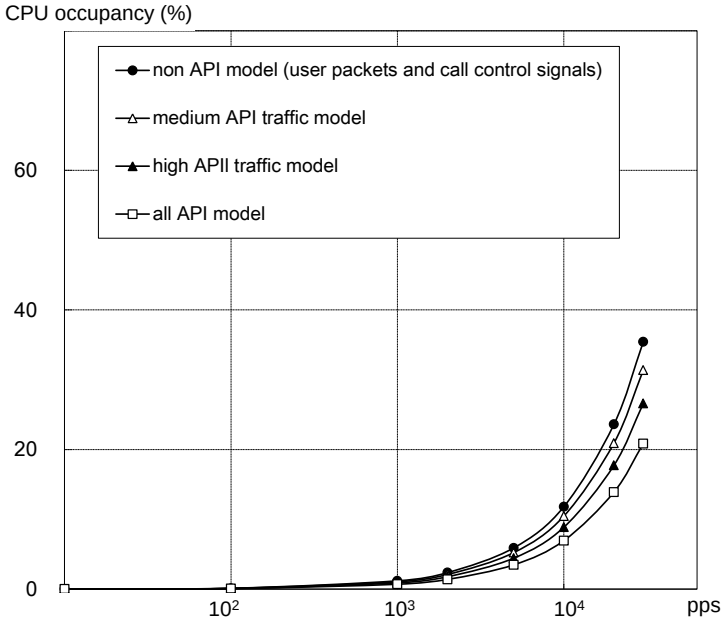


Figure 9. Performance Evaluation Results for Model 3

A first glance at the results, shown in Figure 9, indicates that API request traffic led to a lower CPU occupancy for a given volume of user data packets transferred than was the case with non-API traffic model. This is because API requests do not invoke control signals for communication initiation, which have a large impact on the CPU use ratio. However, there is also a need to note that the extent to which the CPU use ratio is reduced with increasing API requests is equivalent to the reduction in control signals for communication initiation. As the proportion of user packets conveying an API request increases, the total number of call initiations may fall. Thus, for a given volume, a larger volume of API requests removes opportunities for communication requests. This analysis implies that mobile operators should take into account the impact of user packets conveying an API request.

6. Conclusions

This paper has introduced methods for evaluating the performance of a mobile network handling API-oriented traffic. Prior research on traffic analysis in a legacy system can be extended to address one of the major operational challenges in the IMS network with service enablers. We evaluated the impact of estimated API requests on the core network. Traditional indicators, such as packets per second (PPS), are insufficient by themselves because new call control signals and user packets invoked by API requests can have a large impact on the CPU use of the gateway modules. Thus, network capacity planning should take account of the impact of estimated API traffic on a mobile network. Our proposal enables mobile operators to construct a highly stable and reliable system for service innovation by providing APIs to application developers.

In future work, we will clarify the process sequences triggered by an API request and then re-evaluate the performance of the network based on modified API traffic models. It will also be important to develop a method for congestion control using API-oriented traffic models. One possible approach will include distinguishing arriving call packets from service requests

originating from an open API. Commercial API traffic analysis is expected to be a key factor in improving the performance evaluation accuracy and effectiveness of congestion control.

References

[1] Vodafone, http://www.betavine.net/home/main/home.html.

[2] Telefonica, http://www.telefonica.com/en/home/jsp/home.jsp.

[3] Open Mobile Alliance (OMA), "Reference Release Definition for Parlay Service Access. Approved Version 1.0", **(2010)**.

[4] Open Mobile Alliance (OMA), "Next Generation Service Interfaces Requirements. Candidate Version 1.0", **(2010)**.

[5] M. El Barachi, R. Glitho, and R. Dssouli, "Control-level call differentiation in IMS-based 3G core networks", IEEE Network Magazine, vol. 25, no. 1, **(2011)**, pp. 20-28.

[6] S. Pandey, V. Jain, D. Das, V. Planat, and R. Periannan, "Performance study of IMS signaling plane", International Conference on IP Multimedia Subsystem Architecture and Applications, 2007 (IMSAA 2007), **(2007)**, pp. 1-5.

[7] M. Oonuki, K. Kobayashi, K. Nakamura, and S. Kimura, "Special Issue on Mobile Packet Data Communications System, Overview of PDC-P System", NTT DoCoMo Technical Journal, vol. 5, no. 2, **(1997)**, pp. 14-19, (in Japanese).

[8] Telecommunication Technology Committee (TTC), PDC Digital Mobile Communications Network Inter-Node Interface (DMNI) Signaling Method of Mobile Packet Communications System: JJ-70.20, **(2001)**.

[9] M. Hanaoka, S. Kaneshige, N. Hagiya, K. Ohkubo, K. Yakura, and Y. Kikuta, "Special Issue on i-mode Service Network System", NTT DoCoMo Technical Journal, vol. 1, no. 1, **(1999)**, pp. 14-19.

[10] D. Ikeda, "2nd Generation Cellular Networks (PDC-P)", in: H. Esaki, H. Sunahara, and J. Murai (eds.), Broadband Internet Deployment in Japan, pp. 38-43, Ohmsha, Tokyo, **(2008)**.

[11] K. Yoshihara, T. Suzuki, A. Miura, and R. Kawahara, "Evaluation of Congestion Control of the PDC Mobile Packet Data Communication System", IEEE Global Telecommunications Conference 2002 (GLOBECOM 2002), vol. 2, **(2002)**, pp. 1965-1969.

[12] A. Miura, T. Suzuki, K. Yoshihara, K. Sasada, and Y. Kikuta, "Evaluation of the Performance of the Mobile Communications Network Providing Internet Access Service", IEICE Transactions on Communications, vol. E84-B, no. 12, **(2001)**, pp. 3161-3172.

[13] D. Ikeda and A. Miura, "Provision of Paging in a Mobile Packet Data Communication System", 4th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT 2001), **(2001)**, pp.176-180.

[14] A. Miura, N. Shinagawa, F. Ishihara, T. Suzuki, and H. Mochida, "Network Design Based on Network and Traffic Characteristics", 19th International Teletraffic Congress (ITC19), **(2005)**, pp. 819-828.

[15] The 3rd Generation Partnership Project (3GPP), "3GPP Specification series", http://www.3gpp.org.

[16] D. Ikeda, T. Suzuki, and A. Miura, "API-Oriented Traffic Analysis in the IMS/Web 2.0 era", 3rd International Conference on Advanced Communication and Networking (ACN 2011), **(2011)**, pp. 11-18.

Authors

Daizo Ikeda received a B.S. from Sophia University, Tokyo, in 1996 and a M.S. degree from Massachusetts Institute of Technology, Boston in 2004. He joined the Research and Development Division, NTTDOCOMO, Inc., Tokyo, in 1996. From 1996 to 2002 he was engaged in research and development of a mobile packet communicationms network for providing the i-mode service and in 2001 as a special committee member of TTC, his proposal was adopted as part of the PDC-P standards by the Japanese standard body. From 2004 to 2008 he conducted research on architecture design for a mobile service platform, including the design of network APIs. His current research interests includes data mining and large-scale data processing.

Toshihiro Suzuki received B.E. and Ph.D. degrees from the University of Tsukuba, Ibaraki, Japan in 1998 and 2008, respectively. He joined the Research and Development Department, NTTDOCOMO, Inc. in 1998. From 1998 to 2002 he was engaged in research and development of the PDC-P network, which realizes the i-mode service. After this, he was engaged in research on mobility management in moving networks until 2004, and then in mobile ad hoc networks and network virtualization technologies until 2008. His current research interest includes big data mining. He received several awards including the Young Researcher Award from the IEICE in 2006 and the Best Papers Award from the International Conference on e-business and Telecommunication Networks (ICETE) in 2004.

Akira Miura received the B.S. and M.S. degrees from Tokyo Institute of Technology, and a doctorate from Hosei University, Tokyo in 1976, 1978 and 2006, respectively. He joined the Electrical Communications Laboratories, Nippon Telegraph and Telephone Corporation (NTT), Tokyo in 1978. From 1983 to 1987 he was engaged in research and development of a Facsimile Store-and-Forward switching system and a Packet/Circuit Hybrid switching system using Content Addressable Memory (CAM) technology. From 1987 he carried out research and development on Intelligent Networks and worked on developing the Personal Handy phone Service (PHS). In 2000, he joined NTT DOCOMO, Inc. in Yokosuka, Japan. From 2000 to 2002 he was engaged in the development of the i-mode system, and then, to 2006, in research of the next generation Communication Systems and traffic engineering system. In 2008 he joined Kumamoto prefectural University in Kumamoto, Japan and is now the dean of the Faculty of Administration.