

Dynamic Management of QoS Parameters

SeungChoel Mun and Ilsoo Ahn

Network Management Solutions Lab.
Telecommunication Systems Division
Samsung Electronics Co., Korea
e-mail: {scmun, isahn}@samsung.com

Abstract

With the growth of digital technologies in recent years, the border of wired, wireless and data broadcasting network has increasingly disappeared, and integration of the network management system has been in progress to ensure that it can provide a variety of high quality contents efficiently. With a rapid evolution of the network, the QoS management that guarantees the quality of network services has become more important. This paper describes the problems with the static management method of QoS parameters, and presents the concept and algorithm for the dynamic management method. Accordingly, it is possible to add QoS parameters in real time, and the same software component can be used to manage other equipment, in the form of a plug-in, reducing the time and cost of software development. This method has been successfully applied to developing management systems for two different types of network equipment without extra modifications.

Introduction

- QoS management includes
 - Traffic and bandwidth management by differentiating the service level
 - Monitoring and analyzing network status
 - Optimal network configuration
- For management of QoS parameters, Key Performance Indicators (KPI's) should be selected and monitored
- Key Performance Indicators can be selected as
 - Raw performance data
 - Calculated data based on multiple raw data
 - Various forms depending on preference of an operator
- Management of QoS parameters can be categorized into
 - Static management and dynamic management
 - Depending on adaptability to various real-time requirements

1. Introduction

The QoS management has been important in all network systems including the wired and wireless and data network and its efficient and stable management has been issued. The QoS management is referred to as all related technologies that manage traffic and bandwidth by differentiating the service level, and the QoS solution means the effective control and management of the network by monitoring and analyzing generated traffic as well as the increase of the network speed within the limited bandwidth.

Thus, the QoS management is designed to avoid collision and overload when the network is overloaded and to figure out the quality clearness of the requested service in real time. However, as proposed by many recommendations of the network systems, since QoS has complex characteristics due to various technologies, functions and policies, and QoS has multiple technologies in accordance with layer, it is not easy to manage QoS.

To manage QoS efficiently and stably, you should select Key Performance Indicator (KPI) which has direct and serious effects upon network service and monitor the selected KPI continuously. If deterioration of KPI occurs, you should take proper action through an immediate analysis, in real time.

Necessary to manage QoS, KPI can be directly selected from single raw performance data or calculated by counting multiple raw performance data. Additionally, it should be managed in various forms depending on individual preference of the operator and special conditions of the network.

The QoS management can be categorized into static management and dynamic management depending on adaptability to various real-time requirements by the network. This paper details the problem of the static management and the dynamic management for the purpose of improving the static management efficiently.

Static vs. Dynamic Management

- Static management
 - Used for most network management systems
 - Selects the type of KPI at the time of development with hard coding
- Problems with static method
 - Hard to create KPI dynamically
 - Hard-coding should be modified for a new requirement
 - Difficult to re-apply to a different network equipment
 - Not effective in time and cost
 - Hard to support rapidly evolving network
- Dynamic management
 - Designed to define new KPI's dynamically
 - Solve the problems with the static method
 - Need a GUI to manage KPI's dynamically
- Advantages of dynamic method
 - Possible to add a new KPI dynamically
 - Good for managing the network in real-time
 - Easy to re-apply to a new network equipment
 - Reduce the time and cost for development

2. Static QoS Management

The static QoS management method has been used for most network management system, and the KPI which is required to manage QoS is specified and managed in advance at the time of developing system with hard coding, it can cause the following weak points:

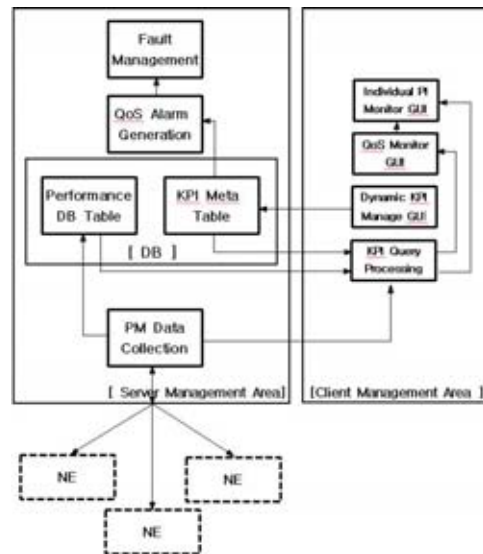
First, KPI to be monitored may vary according to the operator or special conditions of the network; however, it cannot meet the requirements for dynamic KPI creation in accordance with many conditions and operator by selecting KPI in advance when developing the QoS management system. Since hard-coding parts should be modified and redistributed whenever a new requirement occurs, it is not effective in time and cost.

Second, since the static QoS management has an NE tributary KPI-defined algorithm, it is impossible to re-apply it to the different NE management device. Thus, it is not cost-effective because the QoS management device with the same function should be developed repeatedly whenever developing a new NE management device.

As stated above, since the static QoS management method cannot meet the requirements of the network which is rapidly evolved, and it has restrictions in terms of the QoS management, the following section describes the dynamic QoS management method that can overcome its limitation and meet multiple requirements of the network in real time:

Structure of Dynamic QoS Management

- Dynamic QoS management is done with server & client
- Server block
 - Performance data collection module
 - Performance DB table
 - KPI meta table
 - QoS alarm generation part
 - Fault management module
- KPI meta table manages the KPI meta information defined by the operator
- Client block
 - GUI for dynamic KPI management
 - QoS Monitoring Window
 - Individual Monitoring Window



[Figure 1] Dynamic QoS Management Configuration Diagram

APNOM 2005

(4)



3. Dynamic QoS Management

The dynamic QoS management method is designed to define and manage a new KPI dependent by the operator's needs and special conditions of the network even while operating NMS in order to overcome the weak points of the static QoS management.

To create a KPI dynamically, a user interface for defining a new KPI is required, and it is solved through the dynamic KPI management GUI in this paper. The dynamic KPI management GUI enables the operator to select various KPI information so that the operator to add, delete and change a new KPI in real time. As above, since the operator can define KPI anytime through the dynamic KPI management GUI while operating NMS, the dynamic QoS management method can perfectly meet the requirements of the network- which is changing in real-time. And it manages all functions as the form of components. Thus, it is easy to re-apply when developing a new NE management system.

Components required for the dynamic QoS management are as follows:

3.1 Dynamic QoS Management Configuration

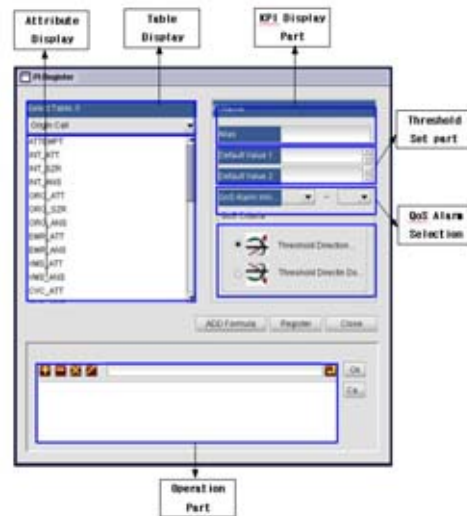
As a part of the performance management function of NMS, the dynamic QoS management is configured as shown in Figure 1 in order to implement the function above. Part 1 indicates the management server of NMS; Part 2 indicates the client management block of NMS; Part 3 is NE.

The client management block is composed of i) a dynamic KPI management GUI that defines a new KPI while operating the network management device, ii) a QoS monitoring window that monitors created KPIs as a whole, and iii) an individual monitoring window that individually monitors KPI, in real time, of which QoS is expected to deteriorate.

The server management block is composed of i) performance data collection part that collects performance data from NE on a regular basis, ii) performance DB table, iii) KPI Meta table that stores and manages the KPI meta information defined as dynamic KPI management GUI, iv) QoS alarm generation part that generates a QoS alarm, and v) Fault Management Part.

GUI for dynamic management of KPI's

- GUI for dynamic management of KPI's
 - Defines KPI Information dynamically
 - Saves the KPI information in the KPI Meta table with JDBC
- Definition of KPI
 - Selection from the raw performance data from equipment
 - Combination of various performance data through arithmetic operation
- Structure of KPI GUI
 - Table Display Part
 - Attribute Display Part
 - KPI Display part
 - Threshold Setting part
 - QoS Alarm Selection part
 - Threshold Direction part
 - Operation part



[Figure 2] GUI for dynamic management of KPI

3.1.1 Dynamic KPI Management GUI

The dynamic KPI management GUI allows the operator to define and add the KPI information to manage QoS dynamically. The definition of KPI can be directly selected from single performance data collected from NE, and can be calculated by the combination of different performance data through operation.

For example, the success ratio of an outgoing call can be KPI as single performance data; but the operator can also define the KPI as the total success ratio of an outgoing call and incoming call.

- Display Part1
Performance data is registered according to the type. If you select the corresponding data, attributes are listed in the list below
- Display Part2
If you select the type of performance data, attributes belonging to the performance data are listed
- KPI Display Part
This part designates names selected by the operator and names used by the dynamic QoS management system
- Threshold Setting Part .
The dynamic QoS management system provides a function to set a threshold so that the operator can, at random, set a reference value of the performance deterioration
- QoS Alarm Selection Part .
The Even if the QoS performance is deteriorated, it is possible to set the restriction period of a QoS alarm so that the operator can prevent a QoS alarm from being generated at random
- Threshold Direction Part .
The This part is used to set the threshold direction of a QoS alarm
- Operation Part .
The If the operator selects KPI, this function allows the operator to select attributes of Display Part 2 and create a formula by the operation. The operator can perform the combination of any operation

KPI Meta Table

- KPI Meta table
 - Stores the KPI information defined through the dynamic GUI.
- KPI Meta table's attribute
 - Ne_id
 - Ne_Title
 - Ne_Type_ID
 - Table_Name: varray Type
 - Table_Title
 - Logical_Name
 - Thres_Formula
 - Thres_Title
 - Thres_value1, Thres_value2
 - Direction
 - Check_value
 - Alarm_Flag
 - From_Hour, ToHour

```

create table_array as VARRAY(10) of
VARCHAR2(50) ; // Object
create logical_array as VARRAY(10) of
VARCHAR2(50) ; // Object
create table pm_threshold
(
  ne_id          NUMBER(5)      NOT NULL
, ne_title      VARCHAR2(50)   NOT NULL
, ne_type_id    NUMBER(2)      NOT NULL
, table_name    table_array    NOT NULL
, table_title   table_array    NOT NULL
, logical_name  logical_array  NOT NULL
, thres_formula VARCHAR2(500)  NOT NULL
, thres_title   VARCHAR2(500)  NOT NULL
, thres_value1 NUMBER(9,2)    NOT NULL
, thres_value2 NUMBER(9,2)    NOT NULL
, direction    CHAR(1)        NOT NULL
, check_value  NUMBER(9,2)    NOT NULL
, alm_flag     CHAR(1)        NOT NULL
, from_hour    NUMBER(2)      NOT NULL
, to_hour      NUMBER(2)      NOT NULL
);

```

[Figure 3] KPI Meta Information Management DB

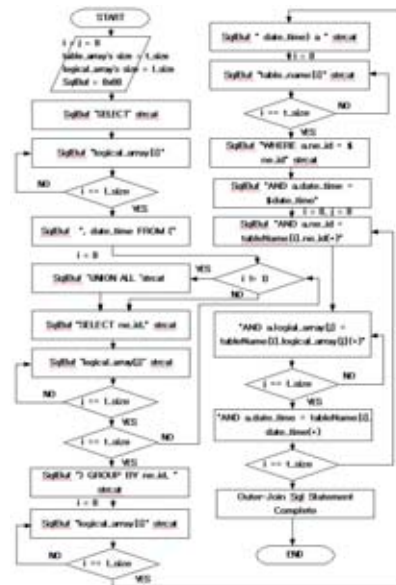
3.1.2 KPI Meta Table

The KPI Meta table is a database table that stores the KPI information defined by an operator. KPI table should contain the network equipment, sub-location, performance database table name, operation formula and threshold information to generate a QoS alarm. Figure 3 shows the KPI DB Table, and the description of Meta information is as follows:

- Ne_ID : Network system ID
- Ne_Title : Network system title
- Ne_Type_ID : Network system type
- Table_Name : The name of the DB table for storing performance data.
- Table_Title : The item name of performance data displayed to the operator mapped to the DB Table name
- Logical_Name : Meta data that stores the Sub-Location information if KPI values are collected as different data in accordance with Sub-Location of the network system.
- Thres_formula : the name of single performance data. If KPI is selected by the operation, it is the formula of operation.
- Thres_Title : KPI name set by the operator
- Thres_Value : The KPI threshold information used to determine whether a QoS alarm is generated or not.
- Direction : Direction of a threshold when a QoS alarm is generated
- check_value : Threshold that is a reference of Critical QoS alarm generated
- Alarm_flag : Information to determine whether a QoS alarm is generated..
- from_hour, to_hour : Time information on preventing a QoS alarm. Even if a QoS alarm is generated while operating the network system, this information is provided so that the operator can ignore a QoS alarm generation.

Query Processing of KPI

- KPI Query Processing Part
 - Creates SQL statement for KPI value
 - Query the current KPI value using the created SQL statement
 - Use outer-join method of SQL
- The processing of Null value
 - For the purpose of correct calculation
 - Null value has to be substituted by '0' or '1'
 - (+)(-) : 0 (*) (/) : 1
- Monitor KPI value through this KPI query processing part



[Figure 4] Dynamic Outer-Join SQL Statement Creation Flow

APNOM 2005

(7)



3.1.3 NE Performance Data Collection Part

To monitor the QoS of the network system, it is necessary to regularly collect the performance data related to the QoS, and store the data in a DB of NMS. The Performance Data Collection Part measures data occurred while operating the network system by the minimum cycle and operator's request, so that the operator can use the measured data for O&M, expecting the demand for equipment and managing the network .

3.1.4 KPI Query Processing Part

The KPI Query Processing Part creates query SQL statement in order to monitor KPI defined by the operator, and queries the current KPI value using the created SQL query statement whenever collecting performance data from NE. This part allows the operator to query the KPI values created from many DB tables through the Outer-JOIN method of SQL .

The Outer-JOIN is used for the case that KPI is calculated value from multiple raw data and some specific data among them are not collected from NE so that the DB table is filled with null. If an event is not generated in the network field, network element does not report the performance data to the management system, considering the NE performance.

An error occurs if you retrieve the formula including null values through the JOIN SQL statement. This part makes it possible to query the KPI values through the Outer-JOIN SQL statement even if specific data is not collected Outer-JOIN SQL. In addition, the performance data (Null value) should be substituted by 0 or 1 for the purpose of correct calculation. In other words, if the value is (+) (-), the performance data should be substituted by 0, and if the value is (*) (/), it should be substituted by 1. In this case, the KPI Query Processing Part performs the operation by replacing 0 or 1 and corrects the error of the KPI value to a maximum. The dynamic QoS management function is used to monitor QoS created through the operation of different data by using the KPI Query Processing Part, and generates a QoS alarm if the current KPI value violates the set threshold

Dynamic Outer-Join Statement

- Why Outer-Join Statement ?
 - To calculate combination KPI's
 - Some data are not collected from NE
 - Then the data is filled with NULL in DB
 - General join statement cannot query the KPI value correctly in this case
- Example of dynamic SQL statement
 - APN Packet Drop Ratio at GGSN
 - : KPI formula
 - $$\frac{APN_STAT_R.uplnk_drop}{APN_STAT_R.uplnk_pkts}$$
 - Route Call Setup Ratio at MSC
 - : KPI formula
 - $$\frac{(CG_INC_R.success + CG_OUT_R.success)}{(CG_INC_R.attempt + CG_OUT_R.attempt)}$$

```

APN Packet Drop Ratio at GGSN
SELECT * FROM
  SELECT apn_stat_r.apn_id,
         to_char(apn_stat_r.date_time, 'YYYY-MM-DD HH24:MI:SS'),
         NVL( apn_stat_r.uplnk_drop, 0)/NVL( apn_stat_r.uplnk_pkts, 1)
  FROM apn_stat_r
 WHERE apn_stat_r.ne_id = 10201 AND apn_stat_r.xtsg_no = '100'
 AND to_char( apn_stat_r.date_time, 'YYYY-MM-DD HH24:MI:SS')
 > '2005-05-01 11:00:00'
 AND to_char( apn_stat_r.date_time, 'YYYY-MM-DD HH24:MI:SS')
 < '2005-05-01 14:00:00'
 ORDER BY apn_stat_r.date_time ;

Route Call Setup Ratio at MSC
SELECT * FROM
  SELECT a.xtsg_no, to_char(a.date_time, 'YYYY-MM-DD HH24:MI:SS'),
         NVL( ca_inc_r.success + ca_out_r.success, 0)
         /NVL( ca_inc_r.attempt + ca_out_r.attempt, 1)
  FROM (SELECT ne_id, xtsg_no, date_time
        FROM (SELECT ne_id, xtsg_no, date_time FROM ca_inc_r UNION ALL
              SELECT ne_id, xtsg_no, date_time FROM ca_out_r)
        GROUP BY ne_id, xtsg_no, date_time) a, ca_inc_r, ca_out_r
 WHERE a.ne_id = ca_inc_r.ne_id(+) AND a.xtsg_no=ca_inc_r.xtsg_no(+)
 AND a.date_time = ca_inc_r.date_time(+) AND a.ne_id = ca_out_r.ne_id(+)
 AND a.xtsg_no = ca_out_r.xtsg_no(+) AND a.date_time = ca_out_r.date_time(+) ) a
 WHERE a.ne_id = 10201 AND a.xtsg_no = '100'
 AND to_char( a.date_time, 'YYYY-MM-DD HH24:MI:SS') > '2005-05-01 11:00:00'
 AND to_char( a.date_time, 'YYYY-MM-DD HH24:MI:SS') < '2005-05-01 14:00:00'
 ORDER BY a.date_time ;
    
```

[Figure 5] Created Outer-Join SQL Statement

※ Example of dynamic SQL statement

1. Example of dynamic simple SQL Statement

QoS Parameter Name : APN Packet Drop Ratio at GGSN System

Description : Uplink Packet Drop Ration according as apn

Performance Table : Apn_Stat_r

Logical Location : apn_id

Formula : $APN_STAT_R.uplnk_drop / APN_STAT_R.uplnk_pkt$

2. Example of dynamic Outer-Join SQL Statement

QoS Parameter Name : Route Call Setup Ratio at MSC

Performance Table : CG_INC_R & CG_OUT_R

Logical Location : xtsg_no (Route number)

Formula : $(CG_INC_R.success + CG_OUT_R.success) / (CG_INC_R.attempt + CG_OUT_R.attempt)$

QoS Monitoring Window

- QoS Monitoring Window
 - Queries the current all KPI value
 - Displays the current QoS status in real time
 - Displays with 3-status color :
Normal / Warning / Critical

- Individual KPI Monitoring Window
 - Displays the past and current KPI value when KPI status is warning
 - Monitors the minimum cycle (5min) data for the past three hours
 - Monitors one hour data for 24 hours
 - Displays the real-time transited value.
 - Forecasts the time of performance degradation (QoS Alarm)



[Figure 6] QoS Monitoring Window



[Figure 7] Individual KPI Monitoring Window

APNOM 2005

(9)



3.1.5 Monitoring Client Area QoS

The monitoring client area QoS retrieves the current status of KPI through the KPI query processing part, and displays the current status to the operator in real time to enable efficient QoS management. It also compares the status with the threshold set in the KPI Meta table to divide the KPI status into Normal, Warning, and Critical in color. If the KPI status is Critical, this part marks the data in red so that the operator can monitor the QoS deterioration in real time

If the status of a specific KPI is Warning, double-clicking on the cell will display the minimum cycle (5 min.) data for the past three hours and one hour data for 24 hours to the operator in the form of chart and table. Whenever the minimum cycle of time is transited, the monitoring client area QoS part displays the KPI flow in real time through graph and expects the time of performance deterioration to prevent the Critical status in advance .

3.1.6 QoS Alarm Detection Part

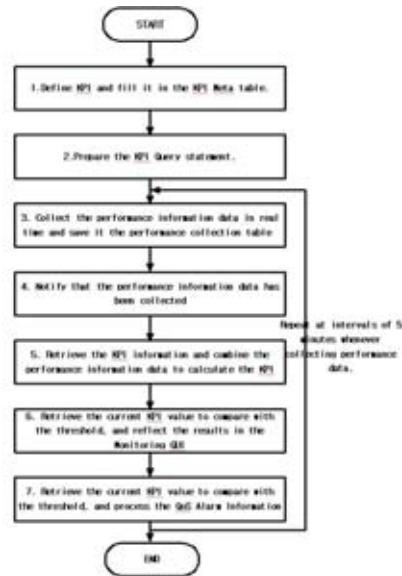
The QoS alarm detection part is a server module that retrieves the current KPI values through the KPI search engine and monitors whether to violate the set threshold. This part generates a QoS alarm and reports it to the fault management part to ensure that the operator can manage it

3.1.7 Fault Management Part

The fault management part saves various QoS alarms reported by the QoS Alarm detection part in a DB and allows the operator to search the alarm history. It also provides. It also provides various functions to manage alarms and faults.

Methodology for Dynamic QoS Management

- Create KPI
 - through the dynamic KPI management GUI
- Prepare KPI query statement
 - Outer-join SQL statement
- Collect and save performance data
- Notify collection completion
 - To Client QoS monitoring GUI
 - To server QoS Alarm detection part
- Retrieve KPI value setting
- Reflect client area QoS
 - Retrieve the current value
 - Compare it with the threshold
- Detect server QoS Alarm



[Figure 8] Dynamic QoS Management Flow Chart

3.2 Dynamic QoS Management Methodology

The entire operation flow of the dynamic QoS management algorithm is as follow:

1) Create KPI

The QoS management device operator creates KPI through the dynamic KPI management GUI and registers it in the KPI Meta table.

2) Prepare KPI Query SQL Statement

The KPI query processing part makes the Outer-Join SQL statement by using the performance data table, location information and KPI formula.

3) Collect/Save Performance Data

The performance data collection part collects performance data from the network system on a regular basis, and saves it in a DB Table.

4) Notify Collection Complete

After collecting data, the performance data collection part notifies the Client QoS monitoring GUI and Server QoS alarm detection part that the performance data is completed collected.

5) Retrieve Setting KPI Information

The KPI query processing part retrieves the KPI information created by the operator in the KPI Meta information table by request of the QoS monitoring GUI and QoS alarm detection part.

6) Reflect Client Area QoS monitoring

The QoS monitoring GUI request the KPI Value Query engine to retrieve the current value of the registered KPI and compare it with the threshold in order to divide into Critical, Warning, and Normal

7) Detect Server QoS Alarm

The QoS Alarm detection part requests the KPI Query engine to retrieve the current value of the registered KPI and compare it with the threshold. If a KPI value violates the threshold, the part notifies a QoS alarm to the fault management part.

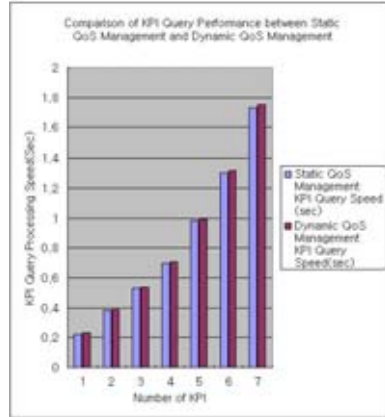
4.4 Communication using HTTP

Applications and Evaluation

- Actual applications for :
 - W-CDMA Core Network management System (CN-EMS)
 - CDMA PTT management System (PTT-EMS)

Number of KPI	Static QoS Management KPI Query Speed (sec)	Dynamic QoS Management KPI Query Speed(sec)
1	0.22	0.23
5	0.38	0.39
10	0.53	0.54
15	0.7	0.71
20	0.99	0.99
30	1.3	1.32
50	1.73	1.75

- Evaluation
 - Almost no difference in terms of KPI query performance (Figure 8)
 - Possible to add & delete the KPI dynamically while operating system.
 - Possible to re-apply the CN-EMS QoS management system to PTT-EMS without changing any code
 - Reduced time and cost required to develop a new application (PTT-EMS)



[Figure 9] Comparison of Static/Dynamic KPI Query performance

4. Application Cases and Evaluation

The QoS management method stated in this paper has been applied to the W-CDMA Core Network management system (hereinafter referred to as W-CDMA CN-EMS) and CDMA PTT management system (hereinafter referred to as PTT-EMS).

When CN-EMS was developed at the early stage, the QoS management was implemented through the static management method, but it was changed to the dynamic management later. In that process, it was possible to compare the performance between the static management and the dynamic management. Figure 9 shows the comparison of time performance between the static management and dynamic management in accordance with the number of KPI. It showed that there is almost no difference between the two management methods in terms of performance because the dynamic management method has no extra process that can deteriorate the performance of QoS management, only except the KPI Meta table query.

On the other hand, the results revealed that the software quality can be upgraded by applying the dynamic QoS management system to the PTT-EMS without changing codes, and it is possible to reduce time and labor required to develop the QoS management system of PTT-EMS.

Conclusions

- Easy to be customized for operators' need dynamically
: possible to create, delete and change the KPI on-line
- Easy to meet requirements of complex QoS management
: possible to create new KPI's as needed
- Reduced development time and cost : possible to be re-used
for different types of equipment

5. Conclusion

This paper described the problems with the static management method of QoS parameters, and presented the concept and algorithm for the dynamic management method with the following benefits.

First, the dynamic management method can create, delete and change the KPI's dynamically without interruption of the existing NMS.

Second, it can create a new KPI by combining various parameters as needed, meeting new requirements of complex QoS management.

Third, the software for dynamic management can be re-used for managing different types of equipment in the form of plug-in, reducing time and cost of new development. This method has been successfully applied to developing management systems for two different types of network equipment without extra modifications.

6. References

- [1] Zhng Wang, Internet QoS: Architectures and Mechanisms for Quality of Service, Morgan Kaufmann Publishers, March 2001
- [2] William C.Hardy, QoS Measurement and Evaluation of Telecommunications Quality of Service, John Wiley & Sons, June 2001
- [3] Paul Ferguson, and Geoff Huston, Quality of Service: Delivering QoS on the Internet and in Corporate Networks, John Wiley & Sons, Inc. 1998
- [4] G. Hwang, D. Lee, J. Jo, J. Kim, S. Jung, and O. Byeon, "Implementation of an on-demand bandwidth-based QoS allocation system for end-to-end resource reservation," in Proc. JCCI'01, 2001
- [5] I. Khalil and T. Braun, "Implementation of a bandwidth broker for dynamic end-to-end resource reservation in outsourced virtual private networks," in Proc. LCN'02, Page 511-519, 2000
- [6] Paul Ferguson, Geoff Huston "Quality of Service Delivering QoS on the Internet and in Corporate Networks", Wiley Computer Publishing, 1988
- [7] Choi Tae-sang, Jung Yun-hee, Son Seung-won, "Trends of Internet QoS Management Technologies", ETRI Weekly Technology Trends, Volume No. 918, pp. 1-22, October 16, 1999
- [8] Trends of Internet QoS Management Technologies <http://stat.ssu.ac.kr/%7Ebatman/Doc/>