A QoS Mapping Rule Builder

K.K. Nguyen[1], B. Kerhervé[2], B. Jaumard[3] *IEEE Member*

[1,3]Concordia University [2]Université du Québec à Montréal

kk_nguye@encs.concordia.ca, bjaumard@ciise.concordia.ca kerherve.brigitte@uqam.ca

Abstract

Although many QoS management architectures have been recently introduced with a lot of advanced features, they have never been widely used in the existing applications due to the lack of interoperation between providers and users, or between network operators. One of the main issues is the heterogeneity of QoS information coming from different sources: clients, communication networks, servers, data .etc. In the context of Quality-Driven Delivery (QDD) referring to the ability of a system to deliver data objects while considering the end-users expectations, all components of a distributed multimedia system have to contribute to satisfy users requirements. The mapping activity is therefore essential for dealing with the variety of QoS information of these components. In this paper, we propose an approach aimed at creating QoS mapping rules using statistical data analysis and data mining techniques combined with monitoring tools. The automatic generation of QoS mapping rules allows adapting the QoS management architectures to different environments as well as different classes of users.

Keywords – QoS mapping, QoS information, Quality-Driven Delivery (QDD), Quality of Service (QoS),

1. INTRODUCTION

Internet applications, especially multimedia services, are demanding huge amount of network and processing resources. They have been widely developed, leading to the need of QoS provision, used to provide flexible adaptation of the system offer to the user requirements. Many QoS management architectures have been introduced in order to deal with this issue [7]. Unfortunately, they are often application-specific and have not been widely used due to the fact that they do not take into account the huge number of QoS information coming from different components of the system: client, communication network, server, data .etc.

One of main challenges for QoS management is the heterogeneity of QoS information. Typically, a QoS architecture is composed of four main layers [6]: the user layer, the service layer, the system layer, and the resource layer. The QoS specified or offered by each layer can be expressed by different abstract levels. For example, QoS specification at the user layer often includes very abstract information such as good, bad, or DVD quality, telephone quality, etc. Lower layers, namely service and system and resource layers, state their offers using more technical details like *frame rate*, *number of color bit*, or *available bandwidth*, *packet rate*, or *CPU throughput*, *memory buffer capacity*. The QoS manager is then responsible to achieve the QoS level corresponding to user requirements (Fig.1). Thus, mapping should be provided to transform QoS information from different layers. In addition, each layer can be composed of a number of components used for a specific application. For example, the service layer can also have a specific DBMS component providing data access for a database-driven application. In such a case, specific QoS information should be taken into account such as *database_response_time* or *number_of_transactions_per_second.* As a consequence, in order to be general-purposed, a QoS management architecture must provide flexible adaptation to the changing conditions of the environment or applications. One of the essential conditions is that the mapping should be processed dynamically.

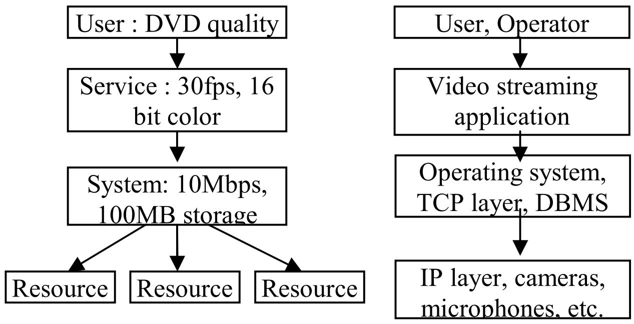

Figure 1: QoS information of layers

This issue led us to propose an approach for implementing a QoS mapping rule builder which is responsible for generating mapping rules from QoS information coming from different sources. The working principle of the QoS mapping rule builder consists in mining the statistical data and configuration files containing working states and configurations of each component of the system, in order to produce association rules describing the QoS relationships between them. Two main techniques are used for this task: i) classification and prediction of user QoS requirement, based on client-side configuration, and ii) clustering system runtime information.

The rest of this paper is organized as follows: Section 2 reviews the QoS mapping activity and presents a typology of mapping rules. In Section 3 we present the monitoring tools used for collecting statistical data and discuss the ability of using statistical data to generate QoS mapping rules. In section 4 we detail the data mining techniques used for this goal. Section 5 concludes and presents future work.

2. QOS MAPPING

Most of current networking systems are based on layering architecture, where each layer can be considered as a black box with specific input/output data. The mapping activity in general and QoS mapping in particular is therefore used to perform the translation of data coming from different layers. Mapping is essential for making QoS decisions within a QoS architecture or transforming information when the system is migrated. We identified two sorts of mapping, vertical and horizontal, used for transforming information between layers and system components [6].

Mapping rules are the basic elements of the mapping activity. A mapping rule transforms one (or a set of) source QoS information to one (or a set of) target QoS information coming from the same or different layers. QoS mapping rules can be classified into two categories [8]:

- function-based mapping rules which are basically mathematical formula and often created using interpolation methods, and
- table-based rules which are characterized by a lookup table, often defined by users or developers, but which can also be built using experimental tests. A lookup table contains a limited number of entries by which a set of output values can be obtained from the input values. An example of table-based mapping rule is presented in [8].

Function-based mapping is characterized by the accuracy and ease of implementation. Traditional QoS dimensions such as delay, jitter and packet loss can be obtained by function-based mapping [2]. Unfortunately, the number of function-based mapping rules is limited since building and validating a mapping function is usually a big challenge. Thus, generally we do not have a sufficient number of function-based rules for processing all the mappings in a system with an unknown number of specific components. This difficulty leads us to use table-based rules as an alternative method. The limitation of table-based mapping comes from the application of such rules: while function-based rules can be used for a wide range of applications or environments (i.e. packet rate calculation can be used in all the TCP/IP applications), table-based rules are only valid within specific working conditions where the environment is supposed to be stable for a long period. However, we consider it as a minor limitation since the QoS problem is often dealt with and characterized by a specific application or system, for example a video streaming delivery system with a defined set of components. This paper focuses on exploiting statistical information in order to build table-based mapping rules for QoS mapping between components of a system.

3. COLLECTING QOS STATISTICAL DATA

One of most important activities in QoS management is collecting QoS information. User and system QoS information are mostly needed. Current QoS-enabled applications usually provide graphical interfaces allowing user to declare their desired quality level and the acceptable cost [3]. On the other hand, QoS information about the runtime environment is collected by monitoring tools which can be classified into three categories [1]:

- The hardware tools integrated into equipments or devices (i.e. router, switch, bridge). These tools can measure particularly the QoS level of the transmission medium or environment, or the performance, etc;
- The software tools integrated into applications. These tools can be used to measure the end-to-end QoS at the application layer;
- The monitoring tools independent of hardware devices or applications. They are developed using specific protocols, such as ICMP, RTCP, SNMP, etc., They can measure some particular QoS parameters of the systems.

Monitoring tools usually record statistical data about the working state of components into *logfiles*. These files, whose size can vary from some megabytes to some gigabytes, contain huge volumes of data values [4] that can be analyzed. Examples of statistical data can be:

- Real-time statistical generation of how content is consumed,
- Ability to analyze data by time of day,
- How many times content was requested,
- Total number of megabytes transferred,
- Costs incurred,
- Quality of transfers,
- etc.

Monitoring tools are often working independently and do not support the aggregation of information in order to build the relationships between different parameters. A few monitoring tools are accompanied by an analyzing module being able to discard out-of-order information. Indeed, most of them return only the raw data and user must extract the appropriate information according to his requirements. Data mining techniques therefore can be used to extract QoS information or relationships between QoS information. Figure 2 shows our implementation architecture of a QoS mapping rule builder taking as input the statistical data from monitoring tools called *Agents*. The *Collector* is used for storing statistical data coming from different source into a database. The *Analyzer* refines the data set. The *Synthetizer* does the classification or clustering and the *Rule Builder* outputs the mapping rules.

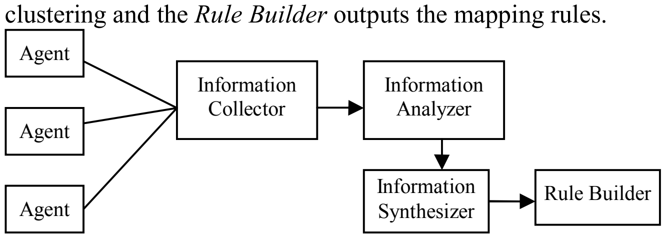

Figure 2: Architecture of the QoS mapping builder

4. MINING QOS MONITORING INFORMATION

This section presents how we apply data mining techniques, namely classification and clustering, in order to build QoS mapping rules from monitoring data.

4.1. Classification of user requirements

One of the highlights of new Internet services is that they often have a huge number of users. In addition, user QoS requirement and devices capabilities are often diverse. Moreover, traditional infrastructures can usually provide only a limited number of quality levels. This issue leads to the need of classifying users and/or user requirements into a limited number of categories so that users can be served with the most appropriate service level. In most cases, users, particularly novice users, cannot describe their requirements in terms of technical metrics such as *frame rate*, *response time* or *loss rate*. Thus, one of the proposed solutions for capturing user requirements consists in showing him a set of samples from which he will choose the desired one [3]. However, this method requires a good cooperation with the user, which is not always available. It can also become unacceptable in case where the user wants to run multiple applications since they can concurrently decrease the device capability so that the test is no more accurate.

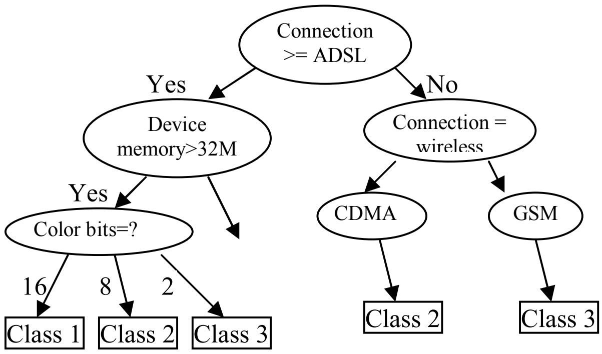

Figure 3: Classification of user requirements

Our classification approach consists in predicting the user maximal capability level (and consequently requirements) based on the configuration information for his device. A decision tree is built, taking into account all the dimensions of the user configuration: connection, memory capacity, display capability, etc. This information can be obtained from monitoring tools and is represented as a node in the decision tree (fig. 2). The ultimate goal is to classify user requirements into groups such as gold, silver or bronze. Building decision tree is basically a greedy algorithm [5], where all possible configurations (or requirements) should be taken into account at each node of the tree.

The classification of user requirements can be performed dynamically at two moments: starting of service or during the service whenever the user configuration is changed. The first moment is used to set up an SLA (*Service Level Agreement*) between user and provider and the later served for realizing a transparent adaptation or renegotiation.

4.2. Clustering system statistical information

Since user requirements may vary and are non-deterministic, we want to predict the class the user belongs to, based on the values of known attributes. So the classification method is deployed as described above. Since the system offers are limited and deterministic, we propose to use the clustering method to get an insight into the data. Note that the system information we mention in this section is not limited to the hardware devices but can cover also the software components and the data itself (i.e. video transmission codec).

In general, QoS dimensions of system, such as bandwidth, vary within a range of values, so called the definition domain. The division of this domain into ranges corresponding to the classes of users requires a clustering method. For example, a router can provide network connections from 1Mbps to 100Mpbs while we only have three classes of users. So we can assign the first class of user to the bandwidth range from 1Mbps to 33Mbps, the second class to the range from 33Mbps to 66 Mbps and so on. However, this division does not take into account the difference of the numbers of users or the diversity of user requirements in each class. Indeed, there are usually more users in the low priority class than in the higher one. User in high priority class has also stricter requirement about QoS level so the bandwidth variation range should be smaller. Thus the definition of service classes corresponding to the user classes in an optimal way is an issue. In this paper, we propose to use the clustering approach to determine the number of levels for a service from the QoS statistical data. The clustering is a set of methodologies for automatic classification of samples into a number of groups using a measure of association, so that the samples in one group are similar and samples belonging to different groups are not similar.

MPEG-2 MOS*	Bandwidth (Mbps)
4.0	5.62
4.1	6.00
4.2	6.47
4.3	7.07
4.4	7.88
4.5	8.99
4.6	10.65
4.7	13.37
4.8	18.64

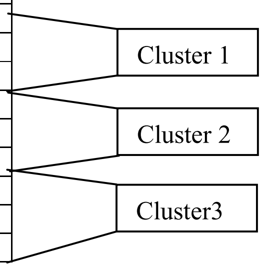

Figure 4: Clustering the system offers

An example of clustering is shown in Figure 4, where:
Cluster 1 = low level bandwidth
Cluster 2 = medium level bandwidth
Cluser 3 = high level bandwidth

In the final mapping rule, these levels can be interpreted as gold, silver and bronze respectively. While the bandwidth varies within a level, a number of transmission codecs with different characteristics is available so that provider can choose the best-fit one to satisfy the user in this class. For example, the user satisfaction corresponding to the cluster 1 can vary from 4.0 to 4.2 in the MOS measuring system. The clustering of

* MOS - *Mean Opinion Score*

system information can simplify the creation of SLA between user and provider. It also allows the provider to change dynamically some parameters (e.g transmission codec) within a level to adapt to user's demand.

The clustering method can be achieved using the K-Means Partitional Clustering algorithm whose performance has been discussed in [5]. Figure 5 presents a result of running the algorithm with a set of statistical data collected from our experimentation where three clusters have been generated.

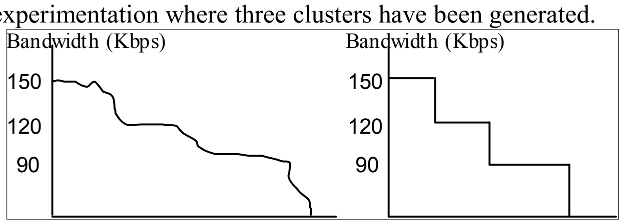

Figure 5: Clustering statistical data

4.3. Building QoS mapping rules

The mapping rule generation process starts with the definition of QoS levels the system is able to provide for a given service. Users will then be classified into categories corresponding to these levels. A mapping rule is an association of user requirement and system offer. It is represented by a table as shown in Figure 6, where a mapping between user classes and system QoS levels is presented. Users belonging to the gold class can be served with the gold QoS level, to the silver class served with the silver QoS level and so on. This mapping is defined based on a set of user parameters and system parameters which we suppose the most important for our experimental service, such as user connection speed, processing power, storage capability, bandwidth, video data MOS, etc. Mapping between less or more parameters is also available. In general, we have three patterns for creating a mapping model:

- One to one mapping: a user parameter corresponds to a system parameter,
- One to many mapping: a user parameter corresponds to several system parameters,
- Many to many mapping: some user parameters corresponds to some system parameters. These parameters must be computed together due to the correlation between them.

In Figure 6, we have a many-to-many mapping rule which transforms a group of user QoS parameters (e.g. connection type, CPU speed, storage capacity) into a set of system QoS parameters and quality of data (e.g. bandwidth and video MOS). Other mapping rules have also been generated, or are being generated over different experiments, allowing to represent the QoS relationships between data network, servers, video documents and different kinds of video players.

Correlation

User class	Connection	CPU (MHz)	Storage (MB)	...	**QoS level**	MOS	Bandwidth (Mpbs)	...
Gold	LAN/T1	>500	>100	...	**Gold**	4.6	18.6	...
Gold	ADSL	>1000	500	...	**Gold**	4.7	13.37	...
Gold	ADSL	>1500	200-500	...	**Gold**	4.8	10.65	...
Silver	CDMA	>300	>30	...	**Silver**	4.5	8.99	...
...	...	...	...	...	**...**	...	...	

User parameters — System parameters

Figure 6: QoS mapping rule

Mapping rules are generated dynamically whenever a new component is added to the system or when the user device or application is changed. The QoS manager is responsible for selecting an optimal mapping rule among the ones that can be used to satisfy user's demand [8].

5. CONCLUSION

In this work we introduced an approach for generating QoS mapping rules from statistical data. We have used two data mining techniques, namely classification and clustering, in order to produce user QoS specification and system QoS offer classes which are then combined into mapping rules. The advantage of our approach consists in the flexibility of the mapping rules generated in terms of components, application, service and environment, particularly relevant for user-oriented QoS architectures. The integration of this approach into a QoS manager is one or our objectives in the future.

6. REFERENCES

[1] Agarwal, D., González, J., Jin, G., Tierney, B., "An Infrastructure for Passive Network Monitoring of Application Data Streams", PAM, 4/2003.

[2] Huard J.-F. and Lazar A. A. "On End-to-End QoS Mapping," Proc. of the Int. Workshop on QoS, pp 303--314, 5/1997.

[3] Jin, J., Nahrstedt, K, "QoS Specification Languages for Distributed Multimedia Applications: A Survey and Taxonomy,", IEEE Multimedia, 2004.

[4] Casale, G. "Combining Queueing Networks and Web Usage Mining Techniques for Web Performance Analysis," 2005 ACM Symposium on Applied Computing, 2005.

[5] Liu, Y., Pisharath, J., Liao, W., Memiki, G., Choudhary, A., Dubey, P., "Performance Evaluation and Characterization of Scalable Data Mining Algorithms," The 16th IASTED Conference, 11/2004, MIT Cambridge, Massachusetts, USA

[6] Nguyen, K., Kerhervé, B., Jaumard, B., "Quality driven delivery for video streaming application: implementation and evaluation", Research Innovation & Vision for the Futur, CanTho, Vietnam, 2/2005.

[7] Schantz, R. E., Loyall, J. P., Rodrigues, C., Schmidt, D. C., Krishnamurthy, Y., Pyarali, I., "Flexible and Adaptive QoS Control for Distributed Real-time and Embedded Middleware," the ACM/IFIP/USENIX Int. Middleware Conf., 6/2003, Brazil.

[8] Kerhervé, B., Nguyen, K., Gebé, O., Jaumard, B., "A Framework for Quality Driven Delivery in Distributed multimedia Systems", ICIW'06 Conference, Guadeloupe, French Caribbean, 2/2006.