# Self-aware Services of NGSDP:Using Bayesian Networks for Measuring Quality of Convergent Services

Zhi Luo, Junping Wang and Qiliang Zhu
School of Software Engineering
Beijing University of Posts and Telecommunications
Beijing, China
luozhi1989@gmail.com, wangjunping@bupt.edu.cn, zhuqiliang@buptnet.edu.com

**Abstract**

We propose a general architecture and implementation for the autonomous assessment of quality of arbitrary service elements in the convergent service environments. We describe a quality engine, which is the central component of our proposed architecture of self-aware convergent services of NGSDP. The quality engine combines domain independent statistical analysis and probabilistic reasoning technology (Bayesian networks) with domain dependent measurement collection and evaluation methods. The resultant probabilistic assessment can be transported via network protocols in the convergent services and it enables non-hierarchical communications about the quality of service elements. We demonstrate the validity of our approach using Multimedia Messaging Service (MMS) Relay/Server and detecting anomalies: storage overflow and message expiration.

**Keywords:** Self-aware, convergent services, NGSDP, MMS.

## 1 Introduction

The telecommunication industry has undergone a great shift from voice centric services towards much more dynamic, convergent services based on IP protocol. The driving force in this revolution has been the introduction of mobile services and a prevalence of the Internet, thus resulting in a change in customer needs. Customer needs can no longer be satisfied by simple services bound to one specific technology. On the contrary, customers expect great flexibility and a complete solution which can deliver end-user services over different technologies. This means that in order to be profitable, telecommunication operators need to transform from simple network access providers to services aggregators, where network access is a vehicle for delivering content centric services.

As the services to be delivered to end users are no longer simple services bound to a single technology, old methods for service fulfillment cease to work. The answer to the integration problems associated with delivering convergent services is leveraging Next Generation Service Delivery Platform (NGSDP) [17]. The main goal for NGSDP is cost effectiveness and reduced time to market for delivering highly customizable bundles of convergent services. System architecture is tuned for delivering services which are aggregations of convergent services rather than monolithic services. The support for multiple play scenarios and delivering services in a value chain is a native functionality of this type of system.

Managing complex hardware and software systems has always been a difficult task. The convergence of services and the proliferation of convergent services have increased the importance of this task, while aggravating the problem in at least four ways:

1. Convergence of services speed software development and release means less reliable and more frequently updated software.

*Journal of Internet Services and Information Security*, volume: 1, number: 1, pp. 46-58

2. Multi-tier and distributed software architectures increase the complexity of the convergent environment and obscure causes of both functional and performance problems.

3. Convergent service construction implies more dynamic dependencies among the distributed software elements of the overall services making it difficult to construct and maintain system models.

4. Convergent services scale deployments increase the number of service elements under a particular administrator's responsibility.

There are some attempts have been made to extend existing management models to include service elements as first-class objects in the research community, such as [13], [5] and [3]. By generalizing Hood's Bayesian network application idea [9, 10, 11] to arbitrary service elements, Alexandre Bronstein [4] has proposed a concept of "Self-aware ness and Control" architecture, which turns out to be a good approach to solve the same problems that exist in Internet-based services. The TeleManagement Forum (TMF) [16] has defined Key Performance Indicators (KPIs) to indicate the performance of the service resources and Key Quality Indicators (KQIs) to indicate the performance of the service elements in wireless services. The KPIs are used to produce KQIs. Our work builds on their successful experiments, by generalizing Bronstein's idea to convergent service elements to get KPIs and use these KPIs to produce the KQIs of the service elements.

The rest of this paper is organized as follows: in section 2, we present the architecture and model for self-aware convergent services of NGSDP, and its properties. In section 3, we describe an instance of that architecture, customized to Multimedia Messaging Services Relay/Server and detection of storage overflow and message expiration. Finally we summarize our contributions and discuss directions for future work.

## 2 Self-aware convergent services of NGSDP

### 2.1 Architecture of self-aware convergent services of NGSDP

This paper proposes the architecture of self-aware convergent services of NGSDP, whose instances can be customized to a wide variety of convergent service elements. We define "self-aware convergent services of NGSDP" as the convergent services in NGSDP whose service elements can autonomously detect deviations in its behavior. We can use this architecture to measure convergent service element's quality and detect whether anomalies exist in it. The architecture of self-aware convergent services of NGSDP is depicted below, in Figure 1.

The Convergent Service Enabler is used to provide the convergence of services. All convergent services will use the functional components and interfaces provided by the framework.

The Convergent Service Enabler provides a framework by defining a horizontal Enabler built on top of network. This framework comprises a set of functional components and interfaces that have been designed to facilitate easy deployment of existing and future convergent services. The components of the enabler are independently reusable [12]. The set of functions interact with one another via the framework provided by the Convergent Service Enabler. The Convergent Service Enabler offers the functionality which can be used to build services.

We propose a quality engine which is deployed in the framework. It draws data from the framework, specifically from each enabler or service element and uses these data to generate opinions about the service element's quality, based on statistical or absolute tests. Then we combine the multiple opinions about the element's quality into a single, probabilistic assessment, using probabilistic reasoning technology. Finally we can communicate the result about the service element's quality to a variety of management applications beyond the top monitor, or to a peer such as a potential customer or supplier

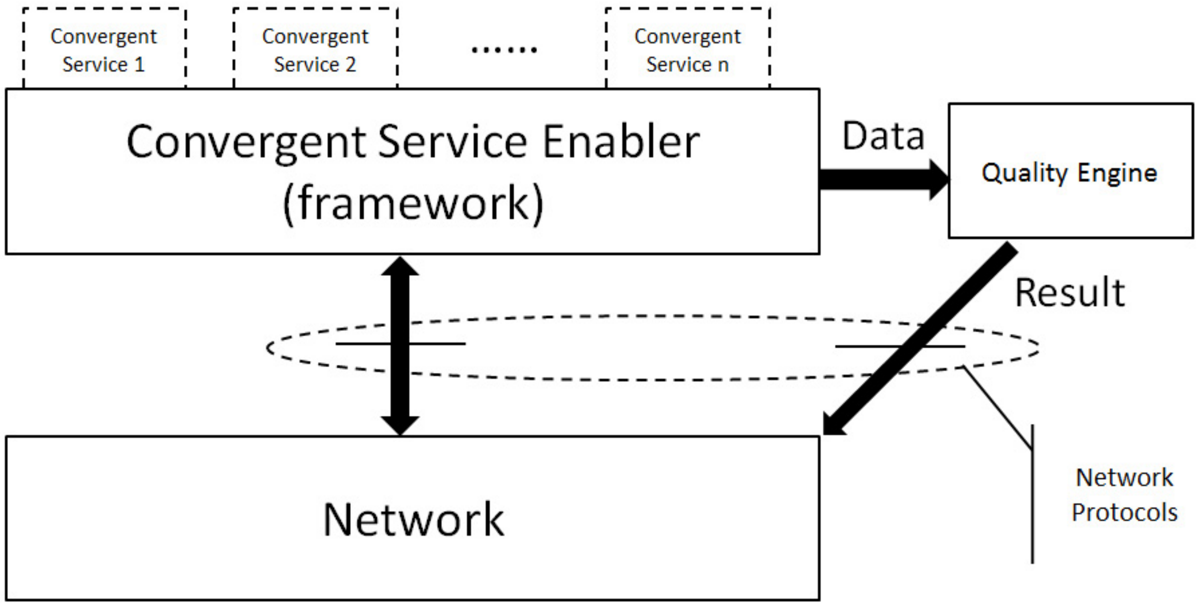



Figure 1: Architecture of self-aware convergent services of NGSDP

of that service element. It can also be passed to the service element itself (Hence the name "Self-Aware Convergent Services"). We transport the result over network protocols and different convergent services may use different network protocols.

### 2.2 Motivation of self-aware convergent services of NGSDP

A convergent service of NGSDP can be viewed as a set of interdependent service elements or objects. Managing such a service, from the point of view of detecting anomalies in its functioning and locating the responsible sub-service elements, is a difficult task. The motivation underlying our work is that such tasks become easier when service elements are aware of their own quality. In the ideal case, all convergent service elements, at all levels of abstraction, are able to accurately assess and efficiently communicate their quality, at all times. Anomaly detection becomes trivial, and fault localization becomes simpler.

Our proposed architecture can achieve this goal by equipping every convergent service element with a quality engine, which can accurately and sensitively evaluates the element's quality. The quality evaluation is based on statistical deviations from past history, so it obviates the need for precise models of the service element's behavior. By using time dependent weighting, the statistical evaluators can adapt to dynamic changes in the service elements.

The quality engine is the central component of our proposed architecture and we will introduce it next in more details.

### 2.3 Quality engine: sensing, evaluating and Bayesian reasoning

The computation of a single, accurate, assessment of quality is key to the expected benefits of this work. We therefore explain the logical model of the quality engine in detail below.

The quality engine achieves its result by first logically wrapping "sensors" around a particular convergent service element. Those sensors generate measurements over time and the measurements are stored in the "measurement packets" in the "measures" layer. We use the terms, sensor and measure, in a broad sense, intended to denote any information capture about the service resources' performance.

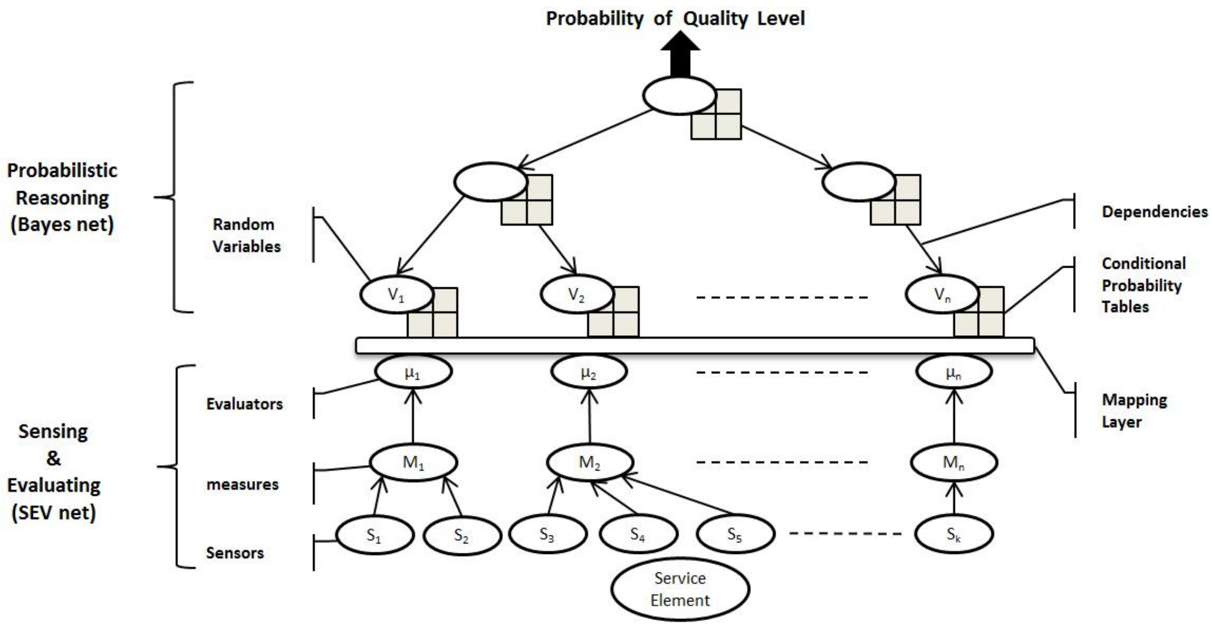



Figure 2: Quality Engine Model

Sensors may or may not require cooperation from the underlying element. Where warranted, sensors that intercept the request/response flow of the underlying element also fit within the architecture.

The second layer in the quality engine accumulates the information provided by the sensors in the measurement packets. The architecture requires that information should be grouped by type to be stored there. The amount of past data to be kept is measure specific. A sensor can only contribute data to one measurement packet but a measurement packet can draw its data from one or more sensors; we denote that information flow with arrows going from the sensors to the measurement packets in Figure 2.

The next layer is the "evaluators". These are functions that process one or more of the measures, to yield a KPI of the service resources' performance. The way of evaluation is set before each evaluator is put to use, including how to process the measures and how to use the processed information to generate a KPI.

The top layer attempts to subtly combine the individual KPIs expressed by the evaluators into a single assessment, using knowledge about the accuracy of the evaluators in different circumstances. In our implementation, we chose Bayesian networks in the top layer for several reasons. Bayesian networks are a proven technology in the field of diagnostics [8] [14] and are capable of leveraging prior expert opinions with learned information from data [7].

The individual quality evaluations are entered as evidence (in the Bayesian reasoning sense) to the Bayesian inference engine. The customized Bayesian networks are specified to have one leaf node ('random variable') for each evaluator. The conditional probability tables encode how much weight to attach to the KPI of each evaluator. Whenever an assessment of the overall quality is needed, the current evaluations are entered as evidence, and the inference engine computes the resulting probability for the top node. The probability is the single assessment of quality level at that instant, which can be used to produce the specific Service KQI of the service element.

# 3 Experimental validation

## 3.1 Storage overflow and message expiration detection using our approach

Multimedia Messaging Service (MMS) is a communications technology developed by the Third Generation Partnership Project (3GPP) that allows users to exchange multimedia communications between capable mobile phones and other devices [1]. MMS can include not just text, but also sound, images and video [2].

The Multimedia Messaging Service Environment (MMSE) is a convergent service environment which provides all the necessary service elements. These service elements may be located within one network or distributed across several networks or network types.

To test the validity of our approach, we applied the approach to the detection of a set of anomalies in Multimedia Message Service Relay/Server, which is the central element in a multimedia messaging service. Messages loss and messages delay are the main problems in a MMS Relay/Server, in this experiment, we focus on the detection of messages loss only, which due to storage overflow and message expiration while messages are temporarily waiting in storage [6].

Figure 3 illustrates our experimental environment, which includes the MMS interface reference points.

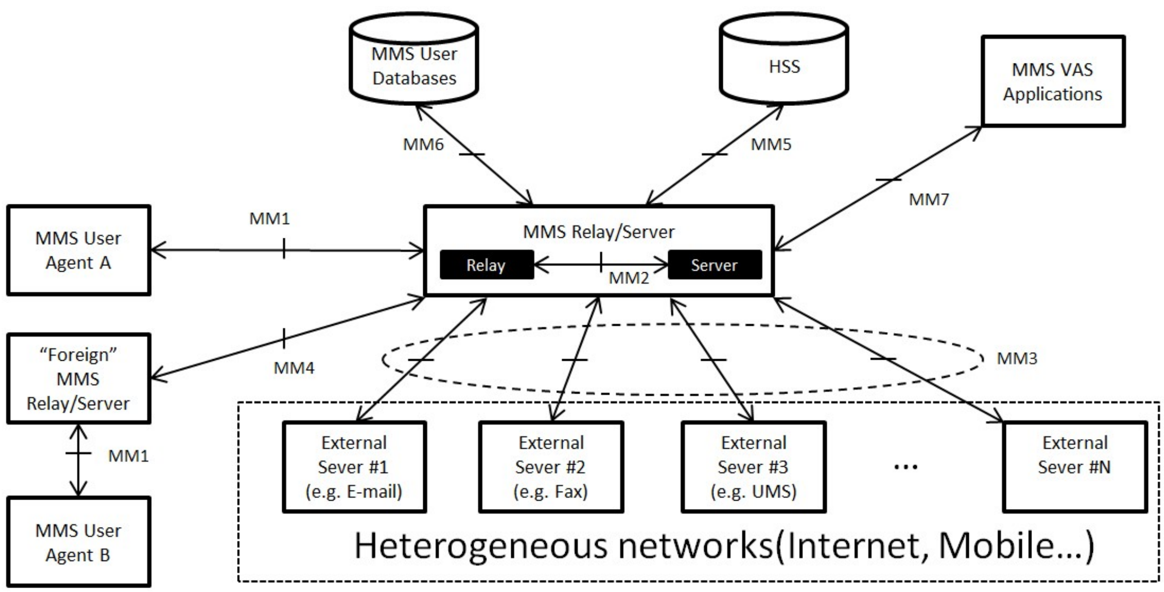



Figure 3: Experimental environment of our approach

## 3.2 The sensors, measures and evaluators

In order to define generic measurements, the 3GPP should specify Performance measurements based on the Reference Points (MM1 to MM7) [15].

The most important Reference Point are the reference points MM1 and MM4, they are the only Reference Point at present that have defined messages that can be used to define Performance Measurements.

Reference point MM1 is used to submit Multimedia Messages from MMS User Agent to MMS Relay/Server, to let the MMS User Agent pull MMs from the MMS Relay/Server, let the MMS Relay/Server push information about MMs to the MMS User Agent as part of an MM notification, and to exchange delivery reports between MMS Relay/Server and MMS User Agents. Following are the measurements proposed by 3GPP:

1. Number of Multimedia Messages submit requests received by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) submit requests received by MMS Relay/Server from MMS User Agent on the Reference point MM1. The measurement name has form MMS.MM1subREQ. We use S1 to represent it.

2. Number of Multimedia Messages submit responses sent by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) submit responses sent by MMS Relay/Server to MMS User Agent on the Reference point MM1. The measurement name has form MMS.MM1subRES. We use S2 to represent it.

3. Number of Multimedia Messages notification requests sent by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) notification requests sent by MMS Relay/Server to MMS User Agent on the Reference point MM1. The measurement name has form MMS.MM1notREQ. We use S3 to represent it.

4. Number of Multimedia Messages notification responses received by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) notification responses received by MMS Relay/Server from MMS User Agent on the Reference point MM1. The measurement name has form MMS.MM1notRES. We use S4 to represent it.

5. Number of Multimedia Messages retrieve requests received by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) retrieve requests received by MMS Relay/Server from MMS User Agent on the Reference point MM1. The measurement name has form MMS.MM1retREQ. We use S5 to represent it.

6. Number of Multimedia Messages retrieve responses sent by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) retrieve responses sent by MMS Relay/Server to MMS User Agent on the Reference point MM1. The measurement name has form MMS.MM1retRES. We use S6 to represent it.

Reference point MM4 between MMS Relay/Servers belonging to different MMSEs is used to transfer messages between them. Following are the measurements proposed by 3GPP:

1. Number of Multimedia Messages forward requests received by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) forward requests received by MMS Relay/Server from another MMS Relay/Server on the Reference point MM4. The measurement name has the form MMS.MM4fwdREQrec. We use S7 to represent it.

2. Number of Multimedia Messages forward responses sent by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) forward responses sent by MMS Relay/Server from another MMS Relay/Server on the Reference point MM4. The measurement name has the form MMS.MM4fwdRESsnt. We use S8 to represent it.

3. Number of Multimedia Messages delivery report requests received by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) delivery report requests received by MMS Relay/Server from another MMS Relay/Server on the Reference point MM4. The measurement name has the form MMS.MM4repREQrec. We use S9 to represent it.

4. Number of Multimedia Messages delivery report responses sent by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) delivery report responses sent by MMS Relay/Server from another MMS Relay/Server on the Reference point MM4. The measurement name has the form MMS.MM4repRESsnt. We use S10 to represent it.

5. Number of Multimedia Messages read reply requests received by MMS Relay/Server. This measurement is a single integer value which provides the number of Multimedia Messages (MM) read reply requests received by MMS Relay/Server from another MMS Relay/Server on the Reference point MM4. The measurement name has the form MMS.MM4readREQrec. We use S11 to represent it.

6. Number of Multimedia Messages read reply responses sent by MMS Relay/Server/ This measurement is a single integer value which provides the number of Multimedia Messages (MM) read reply responses sent by MMS Relay/Server from another MMS Relay/Server on the Reference point MM4. The measurement name has the form MMS.MM4readRESsnt. We use S12 to represent it.

The above twelve measurements are collected by the sensors wrapped around the MMS Relay/Server. Each sensor collects one measurement and every two sensors contribute to one measurement packet in the measures layer. So, we have twelve sensors to collect measurements and six measurement packets to store measurements and each measurement packet store two measurements.

In the evaluators layer, six evaluators are needed and each evaluator processes two measurements which are stored in its corresponding measurement packet. They use every two measurements to generate one measurement, so we get the final six measurements based on the previous twelve measurements:

1. The ratio of MMS.MM1subRES and MMS.MM1subREQ, which is a value between 0 and 1. We use M1 to represent it. The higher ratio is, the less probable messages get lost.

2. The ratio of MMS.MM1notREQ and MMS.MM1notRES, which is a value between 0 and 1. We use M2 to represent it. The higher ratio is, the less probable messages get lost.

3. The ratio of MMS.MM1retRES and MMS.MM1retREQ, which is a value between 0 and 1. We use M3 to represent it. The higher ratio is, the less probable messages get lost.

4. The ratio of MMS.MM4fwdRESrec and MMS.MM4fwdREQsnt, which is a value between 0 and 1. We use M4 to represent it. The higher ratio is, the less probable messages get lost.

5. The ratio of MMS.MM4repRESrec and MMS.MM4repREQsnt, which is a value between 0 and 1. We use M5 to represent it. The higher ratio is, the less probable messages get lost.

6. The ratio of MMS.MM4readRESrec and MMS.MM4readREQsnt, which is a value between 0 and 1. We use M6 to represent it. The higher ratio is, the less probable messages get lost.

All these measurements are observed in the measurement interval, divided by the interval size, 10 minutes in all of these experiments.

Our proposed architecture neither specifies nor restricts the type of evaluation applied to the measurements. In this experiment, we used simple statistical evaluators, namely the mean ($\mu$) and standard deviation ($\sigma$), computed over the last 24 hours, to determine whether the measures indicated a problem. For example, we used $P_i$ to represent the value of M1 measured in the ith measurement interval of last 24 hours ($1 \leq \text{i} \leq 144$). So the mean ($\mu$) and the standard deviation ($\sigma$) of all $P_i$ are determined as follows:

$$\mu = \sum_{i=1}^{144} \frac{P_i}{144}$$

$$\sigma = [\sum_{i=1}^{144} \frac{(P_i - \mu)^2}{144}]^{1/2}$$

It is the same with M2, M3, M4, M5 and M6.

The values of the evaluators for our final six measurements (M1 to M6) were determined as follows:

$$Evaluator = \begin{cases} Good & if\ present\ Measurement \geq \mu + \sigma \\ Medium & if\ \mu \leq present\ Measurement < \mu + \sigma \\ Bad & if\ present\ Measurement < \mu \\ Don't\ Know & if\ no\ past\ measurements\ were\ available\ to\ compute\ \mu\ and\ \sigma \end{cases}$$

### 3.3 The experiment

Figure 4 shows the structure of the system, including the structure of the simple Bayesian network. The twelve sensors correspond to the twelve measurements proposed by the 3GPP which we mentioned above and one sensor collects its corresponded measurement. The six measurement packets store the information come from their corresponded sensors. The six evaluators process the information coming from their corresponded measurement packets to generate final evaluations. Parameters of the conditional probability tables were carefully set, reflecting how much the particular evaluator should be trusted to indicate the presence of the targeted anomaly.

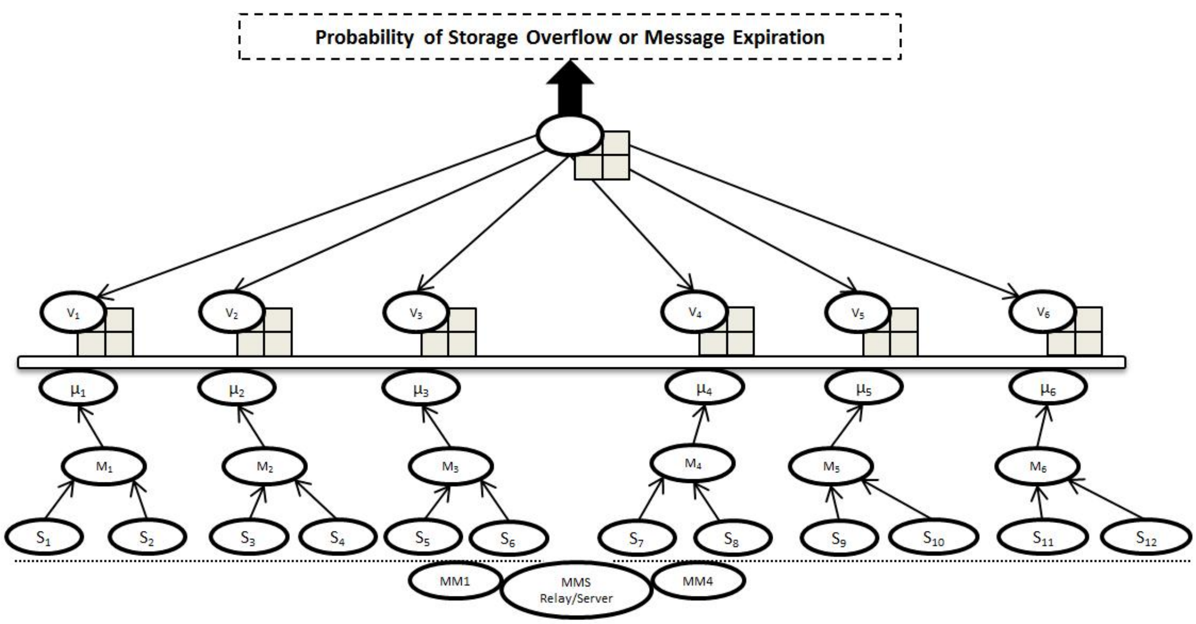



Figure 4: Structure of the Storage Overflow/Message Expiration detector

The experiment we describe was done offline to allow a detailed statistical analysis of the output of the system. We have tested the experiment several times and each time we used the data come from

MMSE of China Mobile Lab Environment to generate the result. After each test, we examined the result and altered the conditional variables to make the reasoning more rational.

### 3.4 Result

The effects of combining the outputs of several evaluators with a Bayesian network are illustrated in Figures 5 and 6. The individual evaluator graphs show both random as well as systematic variability in the individual measurements. Each of the six subfigures plots one of the six defined measurements, as a function of time for one 24-hour period in the experiment.

In each of these sub-figures we have plotted the decision boundaries where the statistical evaluators registered “medium” or “bad”. In sub-figures (a)-(f), corresponding to the six measures, the dashed lines correspond to the time-averaged means and time-averaged means plus one standard deviation, which computed over last 24 hours.

In Figure 6, we display the output of the Bayesian network, corresponding to the probability of the storage overflow or message expiration over the same 24-hour measurement period.

From the six sub-figures in Figure 5, we can see all the six ratios are below the “bad” line at 07:12, which means the performance of the MMS Relay/Server is not good and many messages lost at that measurement interval. At that time point, in Figure 6, we can see the probability of storage overflow or message expiration is very high, which is almost equal to 100%. At another time point, about 15:18, we can see the probability of storage overflow or message expiration is very low from Figure 6, which is near to 0. At that time point, from Figure 5, we can see the ratio of MMS.MM1subRES and MMS.MM1subREQ, the ratio of MMS.MM1notREQ and MMS.MM1notRES and the ratio of MMS.MM4repRESrec and MMS.MM4repREQsnt are above the “medium” line; the ratio of MMS.MM1retRES and MMS.MM1retREQ and the ratio of MMS.MM4fwdRESrec and MMS.MM4fwdREQsnt are between the “medium” and the “bad” line and the remain one ratio is below the “bad” line.

We can use these probabilities as bases to generate the KQIs of the MMS Relay/Server of the corresponded measurement intervals and in this experiment we define the quality level as bad if the probability of message loss is higher than 70%. So we can see from Figures 5 and 6, if four or more ratios are below the “bad” line, the quality level of that measurement interval is tend to be bad. The quality level is good if the probability of storage overflow or message expiration is lower than 30% and from the Figures 5 and 6, we can see if three or more ratios are above the “medium” line, the quality level of that measurement interval tends to be good. The quality level is medium if the probability of storage overflow or message expiration is between 30% and 70%.

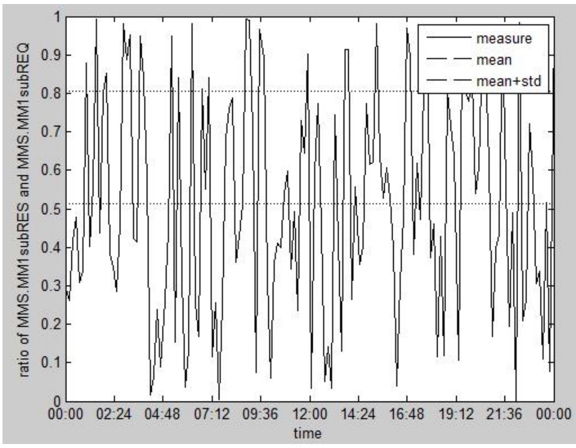
This is work in progress. An attempt is under way to define evaluators better suited to detecting other problems exit in MMS Relay/Server such as messages delay. We plan to publish those results in a forthcoming technical report.

## 4 Conclusion

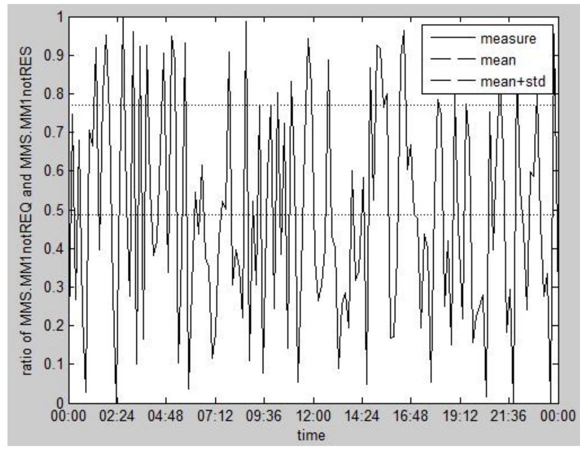
We have presented a general architecture which aims at autonomously assessing the quality of convergent service elements, by adding a layer of intelligence on top of the measurement gathering and sending paradigm. That intelligence is provided by a combination of common statistical techniques, packaged in a reusable way, and a probabilistic reasoning technology (Bayesian networks).

In addition, by using network protocols to transport it, we enable easy peer-to-peer conversations about the service element’s quality. This is a broader paradigm than the hierarchical, client-server model.

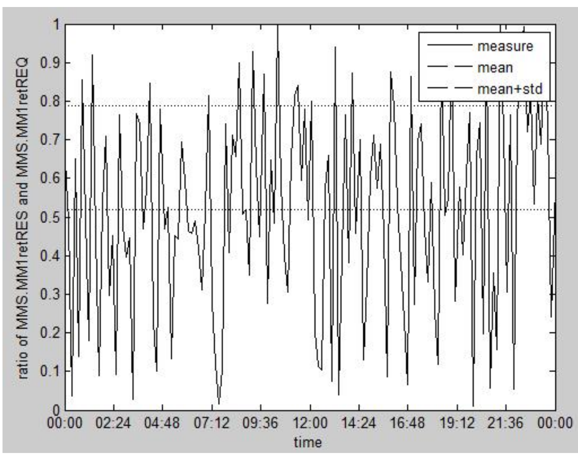
We have experimented with an instance of this architecture, customized to the Multimedia Messaging Services Relay/Server, and targeted at the problems of storage overflow and message expiration. The



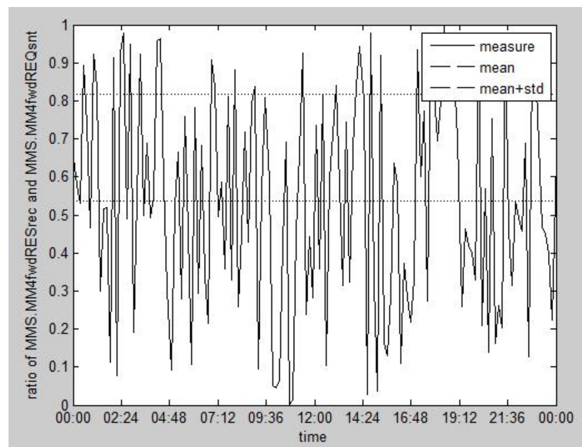

(a) MMS.MM1subRES/MMS.MM1subREQ



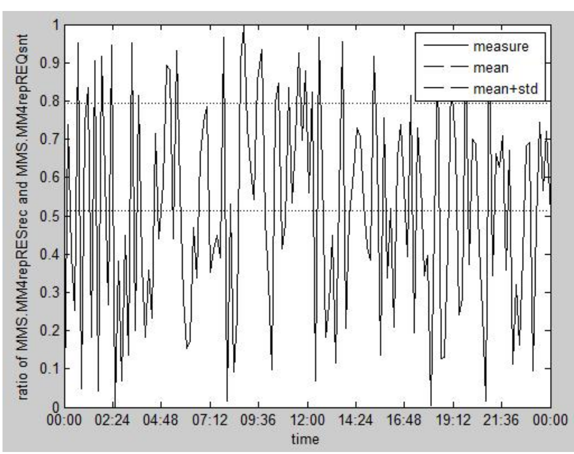

(b) MMS.MM1notREQ/MMS.MM1notRES



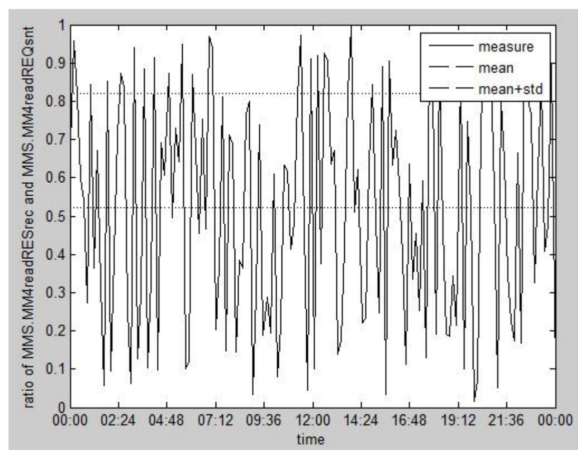

(c) MMS.MM1retRES/MMS.MM1retREQ




(d) MMS.MM4fwdRESrec/MMS.MM4fwdREQsnt




(e) MMS.MM4repRESrec/MMS.MM4repREQsnt




(f) MMS.MM4readRESrec/MMS.MM4readREQsnt

Figure 5: Plot of the measures for all 10 minutes periods of one particular day with the decision boundaries of the evaluators

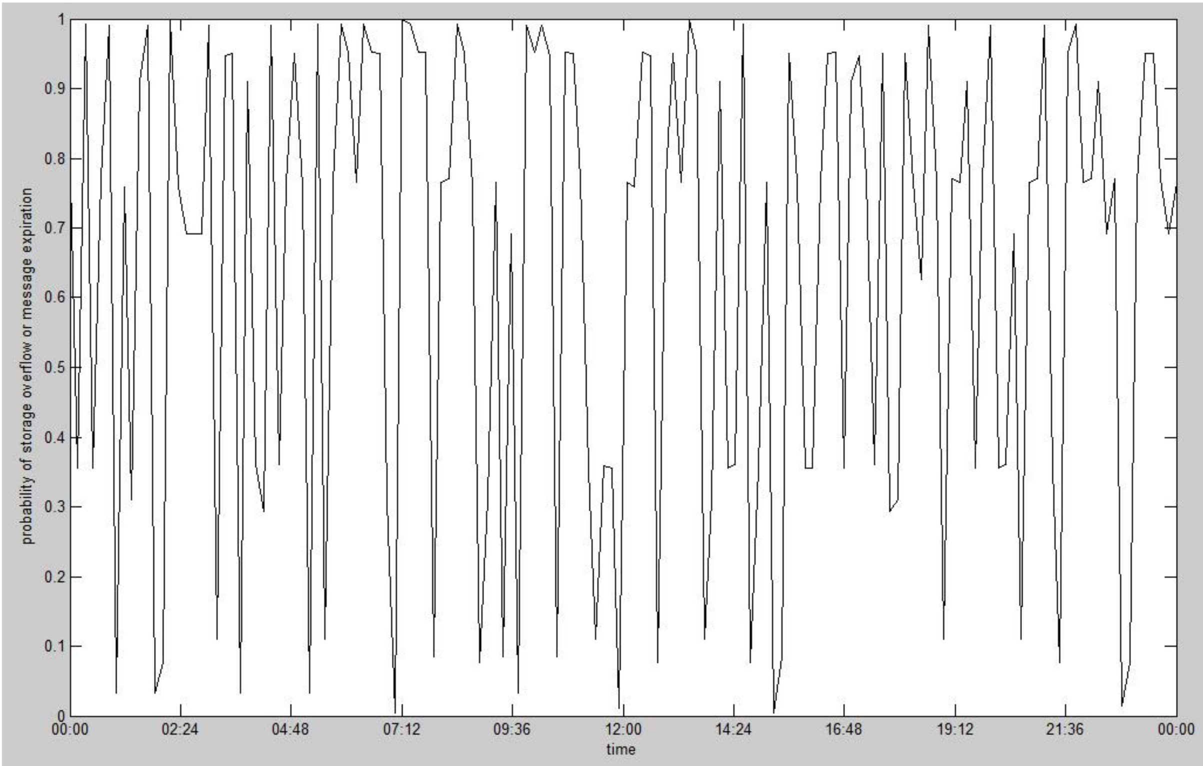



Figure 6: Probability of storage overflow or message expiration, computed over the same 24 hour period as Figure 5

experiment shows encouraging results despite the absence of specificity and sophistication of the sensors, measures and evaluators.

We believe that our proposed architecture is a promising step toward the challenges of managing convergent services. The approach is general and valid for arbitrary convergent service elements. The absence of a requirement for a detailed and complete model of correct behavior is an attractive aspect of this approach. The prospect of greater sensitivity and accuracy by the combination of statistics and probabilistic reasoning is compelling. The ability to reduce a potentially broad and diverse set of noisy inputs to a single number is another advantage of the approach.

### 4.1 Future work

To realize the full potential of these concepts requires more research. We intend to extend this work in a number of ways as well as additional experiments involving customizations of our proposed architecture to various domains. We are researching the application of more sophisticated statistical and probabilistic reasoning technology such as learning, to make the quality engine more accurate.

## References

[1] Definition for multimedia messaging service. http://searchmobilecomputing.techtarget.com/definition/Multimedia-Messaging-Service, Accessed February 2011.

[2] What is mms. http://www.mobile-phones-uk.org.uk/mms.htm, Accessed February 2011.

[3] P. Bhoj, S. Singhal, and S. Chutani. SLA management in federated environment. In *Proc. of the 6th IFIP/IEEE International Symosium on Integrated Network Management, Boston (IM'99), Boston, Massachusetts, USA*, pages 292–308. IEEE, May 1999.

[4] Re Bronstein, Joydip Das, Marsha Duro, Rich Friedrich, Gary Kleyner, Martin Mueller, Sharad Singhal, and Ira Cohen. Self-aware services: Using bayesian networks for detecting anomalies in internet-based services. In *Proc. of the 2001 IEEE/IFIP International Symposium on Integrated Network Management (IM'01), Seattle, Washington, USA*, pages 623–638. IEEE, May 2001.

[5] Deborah Caswell and Srinivas Ramanathan. Using service models for management of internet services. *IEEE Journal on Selected Areas in Communications*, 18(5):686–701, May 2000.

[6] Majid Ghaderi and Srinivasan Keshav. Multimedia messaging service: System description and performance analysis. In *Proc. of the 1st International Conference on Wireless Internet (WICON'05), Budapest, Hungary*, pages 198–205. IEEE, July 2005.

[7] David Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, March 1996.

[8] David Heckerman, Abe Mamdani, and Michael P. Wellman. Real-world applications of bayesian networks. *Communications of the ACM*, 38(3), March 1995.

[9] Cynthia Steiz Hood. *"Intelligent detection for fault management of communication networks"*. PhD thesis, Rensselaer Polytechnic Institute, June 1998.

[10] Cynthia Steiz Hood and Chuanyi Ji. Proactive network fault detection. In *Proc. of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97), Kobe, Japan*, volume 3, pages 1147–1155. IEEE, April 1997.

[11] Cynthia Steiz Hood and Chuanyi Ji. Intelligent agents for proactive network fault detection. *IEEE Journal on Internet Computing*, 2:65–72, April 1998.

[12] Open Mobile Alliance. Convergent IP Messaging Architecture, October 2010.

[13] Srinivas Ramanathan, Deborah Caswell, M. Asawa, and R. Malpani. Service model for email services at Sprint. HP Labs ISAL. Palo Alto, CA, 1997.

[14] Claus Skaanning, Finn V. Jensen, and Uffe Kjærulff. Printer troubleshooting using bayesian networks. In *Proc. of the 13th international conference on Industrial and engineering applications of artificial intelligence and expert systems: Intelligent problem solving: methodologies and approaches (IEA/AIE'00), New Orleans, Louisiana, USA, LNCS*, volume 1821, pages 587–640. Springer-Verlag, June 2000.

[15] TeleManagement Forum. GB923-B Wireless Service Measurements 3GPP Submissions, March 2004.

[16] TeleManagement Forum. GB923 Wireless Service Measurements Handbook, March 2004.

[17] Łukasz Mendyk. Next Generation Service Delivery Platform. White Paper of COMARCH. http://knowledgecenter.comarch.com/en/?id=56, January 2009.

**Zhi Luo** is a college student of Software Engineering in Beijing University of Posts and Telecommunications, China. His research interests include Intelligent and Service Provider technology of trusted distributed network.

**Junping Wang** received the M.S. degrees in Software Engineering from Beijing University of Posts and telecommunications, and joined Beijing University of Posts and Telecoms. He has 6 years research experience in the area of information and communications. His research interests include Service-Oriented Mobile Computing, Intelligent and Service Provider technology of trusted distributed network.

**Qiliang Zhu** graduated from Mockow University of Information and Communications Technology, and joined Beijing University of Posts and Telecoms. He has 40 years teaching and research experience in the area of information and communications. He is the professor of BUPT and CTO of Network Operation & Maintenance Committee of China Communication Enterprise Association.