MDFM: Multi-Domain Fault Management for Internet Services*

Xiaohui Huang, Shihong Zou, Wendong Wang, Shiduan Cheng

State Key Lab of Networking and Switching
Beijing University of Posts and Telecommunications
Beijing, P.R.China, 100876
{hxiaohui, zoush, wdwang, chsd}@bupt.edu.cn

Abstract. New requirements of service-oriented fault management are analyzed and a framework MDFM (Multi-Domain Fault Manager) is proposed in this paper to solve the service fault localization problem in multi-domain context. Different from current solutions, our approach decomposes SLS (Service Level Specification) based on network capability, and monitor service performance in each domain along the end-to-end path. As a result, MDFM can localize the approximate domain rapidly on which the root cause resides, therefore causative region is narrowed down and computation cost for fault analysis is reduced. Faults on both server and client sides are considered in MDFM. A prototype has been implemented to prove the feasibility and efficiency of our service fault management framework.

1 Introduction & Motivation

As Internet migrates gradually to a Service Oriented Architecture (SOA), Service Providers (SP) find out that Internet service has the potential to bring great profits. Thus various Internet services appear, such as Video on Demand, IP TV, VoIP and other multimedia services. In order to maintain regular customers and attract new users, it's necessary for SPs to provide QoS (Quality of Service) for their services.

Fault management is crucial for service QoS guarantee. Service unavailability or performance degradation may cause SLA violation. Therefore, SP desires for a service fault management mechanism, which can perform fault localization and adopt countermeasures as quickly as possible to reduce the service down time and performance degradation period.

In general, an Internet service scenario comprises three parts: server/server farm, client and the network. Consequently, service unavailability or performance degradation may be caused by the faults in the server side, client side or network. Service-oriented fault management should take all these aspects into account, and justify who will be responsible for the service failure.

* This work was supported by the National Basic Research Program of China (Grant No. 2003CB314806 and 2006CB701306), the National Natural Science Foundation of China (No. 90204003 and 60472067) and the National 863 Program of China (No.2003AA121220)

Current Internet services are un-managed or managed by SPs in a non-standard and proprietary way [6]. The private service management can merely deal with the intranet failures due to the lack of network related information. With regard to the network part, present network is divided into several domains (or Autonomous Systems, AS) and belongs to different Network Providers (NPs). End users reside in different geographical regions and most modern services may delegate parts of capabilities to other services distributed in the network. Therefore, service traffic may span several domains.

Multi-domain environment incurs a number of problems for fault management: Different NPs have proprietary network fault management systems without open interfaces for others to retrieve precise information. Thus a certain network element failure is only visible for the NP who governs the corresponding domain, though the failure may propagate to other domains and incur a number of alarms there. Intuitively, fault analysis should be performed in all domains to locate the root cause. However, such method is time-consuming. Before a fault analysis process is initiated, it is necessary to narrow down the causative region so as to shorten the time and improve the accuracy of fault diagnosis.

Therefore, it is necessary to build a universal fault management framework capable of end-to-end fault diagnosis for commercial operation of Internet services in multi-domain context.

Kong et al [6] suggested that the approaches taken in telecommunication industry offer a sound framework for defining Internet service management. However, there are still significant differences between service fault management and traditional network fault management:

- Stable Scenario vs. Dynamic Scenario: Network fault management considers network elements, thus it has a relatively stable view of the network. However, services face the dynamic management scenarios. Due to the distribution of service subscribers and the frequently changing routing information, different services traverse dissimilar network nodes and links, thus have diverse network topologies and management scenarios, which is also mentioned in [1].
- Whole View vs. Partial View: Network fault management has a whole view of all the failures in the network. While in service fault management, failures outside the end-to-end path of the service ought not to be observed by the service manager, even if they affect the service performance.
- Different Alarm Types: Network element malfunction is the major alarm type in traditional network fault management. But in SOA, performance degradation is also a symptom which needs to initiate a fault diagnosis process.
- Different Layers involved: In traditional network fault management, fault diagnosis is focused on lower layers (physical and data link layers). But the fault diagnosis in SOA reaches through to application layer.

In order to fulfill the new requirements of service-oriented fault management, MDFM (Multi-Domain Fault Manager) is proposed in this paper, providing a sound framework for multi-domain service fault management. By SLS (Service Level Specification) decomposition and monitoring, causative region is narrowed down rapidly to a certain domain. In addition, MDFM takes both the end systems and the network into account. A prototype is implemented and proves that MDFM can locate the root cause quickly to prevent service performance from degradation.

The rest of the paper is organized as follows: Section 2 gives a brief overview of current research work related to service fault management. We elaborate on MDFM in section 3. The prototype of MDFM along with the experiment results is presented in section 4. The whole paper is concluded in section 5 with the future work.

2 Related Work

Most solutions from Industry focus on the service fault management inside SP's intranet region. It may attribute to the absence of network related information. HP (Hewlett Packard) laboratory brought forward a series of solutions for Internet service fault management: Darst and Ramanathan [7] proposed a methodology and measurement instrumentation for managing the end-to-end ISP service performance. Caswell and Ramanathan [8] proposed to use service model for Internet Service health management. Bayesian network is employed by Alexandre et al [9] to assess the overall health of the service and to detect anomalies. Besides HP, IBM releases his business service management product, Tivoli [10], which can perform event correlation across multiple environments to identify the root cause of the problem. These solutions concentrate on the service management of SP's intranet, servers, applications and infrastructure components. They are insufficient because they fall short of addressing the network impact on services. In practical situation, faults of network elements along the end to end path have a great impact on services' performance.

In academia, the focus is on service fault localization algorithms and methodology. Andreas and his partners propose service oriented event correlation in [5] and demonstrate its importance in service fault management. However, reference [5] also merely focuses on intra-provider resource, and service fault management is performed within SP's intranet region.

Steinder and Sethi [2] point out that end-to-end service fault localization is a necessity for service management. In [3], a belief network is used as a probabilistic fault propagation model (FPM), and Bayesian reasoning technique is applied to perform fault localization. Steinder and Sethi extend their previous work in [4] and propose a distributed fault-localization technique to solve the end-to-end service fault localization in multi-domain environment. By contrast with aforementioned solutions, network faults are the major concern of [4] and no failures on the server side or the client side are considered. Furthermore, as pointed out by Steinder, accurate propagation patterns and relationships among network events are difficult to be obtained and maintained. Thus in [4], fault localization must be performed in all domains to find out the most likely hypothesis, which is time-consuming and inefficient in practical management system.

3 Multi-Domain Fault Manager – MDFM

In the following we present our Multi-Domain Fault Manager (MDFM), which can fulfill the requirements of service fault management and solve the fault localization problem in multi-domain context. The framework of MDFM (Multi-Domain Fault Manager) is illustrated in Fig. 1, comprising Agents on both the server and the client side, and Fault Analyzer corresponding to each domain (composed of Alarm Collector, SLS Monitor and Fault Diagnostic Toolkit). SLS Decomposer and Monitoring Task Generator depicted in Fig. 2 are outside the Fault Analyzer, yet provide fundamental support for service fault diagnosis, thus they are also regarded as important parts of MDFM.

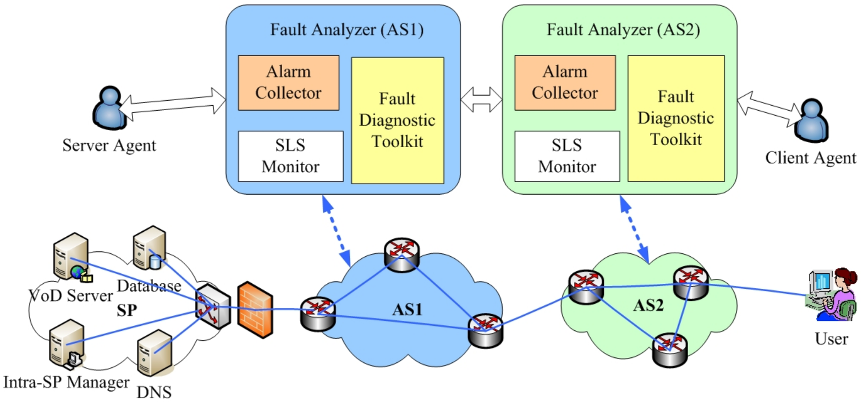


Fig.1. MDFM Framework

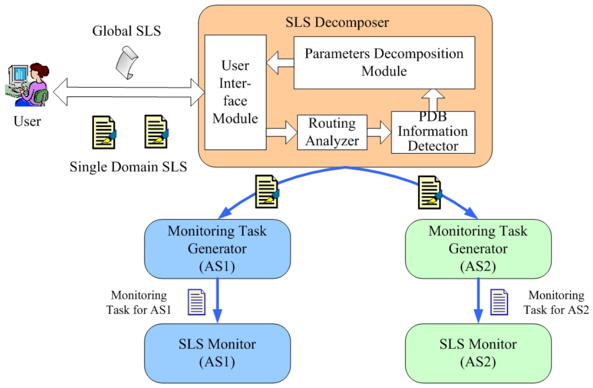


Fig. 2. SLS Decomposer

MDFM is a portion of our QoS provisioning architecture, QoSJava [13], which can provide an end-to-end QoS for users in IP network with heterogeneous QoS mechanisms and network devices. MDFM is responsible for finding out the root cause of service unavailability or performance degradation, and remedying the situation as quickly as possible to guarantee the QoS commitment. User’s QoS requirement is specified in an SLS (Service Level Specification), which is the technical part of SLA (Service Level Agreement). If the subscribed SLS is admitted, monitoring tasks are generated and network devices are configured. When a service failure is reported to Alarm Collector, Agents and Fault Analyzers of all domains along the end-to-end path cooperate to find out the root cause. The capability of each component and the whole process of fault localization are described below.

3.1 Agents on Server Side and Client Side

Internet service is hosted in a server or server farm. Faults may occur in server side, including hardware malfunction, unavailable sub services, excess resource utilization resulted from huge amount of request attempts and so on. All these factors may cause service unavailability or performance degradation on the server side. In these cases, SP bears responsibility for SLA violation.

Users use their computers to obtain services. Service provisioning on the client side needs proper functioning of all layers in the protocol stack. Thus Service performance perceived by an end user is also affected by the performance of his computer. When the user executes too many programs, which devour CPU and memory resource, the speed of the computer is lowered down, and the service performance may degrade subsequently. Sometimes the client side has a narrowband connection, but the user subscribes a broadband service. Yet the user may still complain to SP about the service, though actually he is accountable for it.

Therefore, status of both sides needs to be monitored. Agents resided on both sides collect this information, including CPU usage, memory consumption, and number of open files or run state of a process or a thread. Status of the service can be gained by mapping processes and threads to applications [11]. Agent is an entity which reports service status about the server farm or client host. Our framework places no restriction on agent’s implementation. In fact, SP can build up his own service management system for internal service fault management, and only needs to provide interfaces for our Fault Analyzer to collect the information of service status. Therefore, current solutions of HP, IBM and other enterprises can be integrated into MDFM. In our prototype, agent seems like a task manager of Windows with extended capabilities.

3.2 SLS Decomposer

When subscribing a service, user needs to sign a SLA with the SP. SLS is the technical part of SLA. User’s QoS requirements of Internet service is depicted in SLS in terms of technical parameters quantifying network capabilities. Modern Internet service may delegate some capabilities to other services. In this case, service provisioning involves multiple SPs, which makes the situation complicated. In fact, the service which delegate capabilities can also be regarded as a service subscriber. Thus the scenario is divided into several sub-scenarios consisting of a single SP and a single customer. SLS is signed between each SP-customer pair. We focus on the sub-scenario and call the end-to-end SLS as a global SLS, its formal definition is given by the following tuple:

$$QoSReq \triangleq (SrcIP, DesIP, BW, Class, Delay, LossRate, Jitter, StartTime, EndTime)$$

$$GlobalSLS \in QoSReq$$

The parameters contained in the tuple can be extended as needed. At present, the following items are defined: Source IP Address ($SrcIP$), Destination IP Address ($DesIP$), Bandwidth required (BW), Traffic class of the service ($Class$), End-to-end delay ($Delay$), End-to-end packet loss rate ($LossRate$), End-to-end jitter ($Jitter$), the time when the contract begins to take effect ($StartTime$), and the time when the contract begins to expire ($EndTime$).

Global SLS may traverse several network domains with different capabilities. Thus we propose a PDB (Per-Domain Behavior) based SLS decomposition technique in [14]. PDB [12] is a term from Diffserv and is used here as a representation of network capability, such as delay, packet loss rate and jitter between edge-router pairs in a certain domain. PDB information can be obtained from monitoring statistics. Based on the PDB information, $GlobalSLS$ is decomposed into several SLSs corresponding to each domain along the end-to-end path, which are called single domain SLS. For example, the end-to-end path crosses m domains, then $GlobalSLS$ is decomposed into m single domain SLSs, $SLS_i \in QoSReq \;\; (i = 1, 2, ...m)$, in which SLS_i corresponds to domain i .

If domain i has sufficient resource, SLS_i will be admitted by domain i . $GlobalSLS$ is admitted iff all SLS_i in the end-to-end path is admitted. Thus far, user's QoS requirement is divided among network domains according to their network capabilities and each domain is responsible for fulfilling its commitment.

3.3 Monitoring Task Generator

If a global SLS is admitted, Monitoring Task Generators (MTG) will generate corresponding monitoring tasks for all single domain SLSs in the end-to-end path. Because different network domains may adopt heterogeneous QoS mechanisms and equipped with diverse network devices, which lead to different monitoring methods, MTG generates monitoring task for a domain according to its QoS mechanism. $TaskGen$ is a function mapping a single domain SLS SLS_i to a monitoring task T_i to be performed in domain i . A monitoring task generation process for DiffServ domain is described as follows, and the parameters' semantics are listed in Table 1:

$$TaskGen : SLS_i \to T_i$$

$$T_i \triangleq (Address, Class, MonTime, Param) \qquad Address \triangleq (SrcIP, SrcPort, DesIP, DesPort)$$

$$Class \triangleq Gold \mid Silver \mid Bronze \qquad MonTime \triangleq (StartTime, EndTime, Interval)$$

$$Param \triangleq (Throughput, Delay, Jitter, LossRate)$$

As presented in Fig. 2, SLS monitoring tasks will be deployed in SLS Monitors of all domains along the end-to-end path. SLS Monitor in a certain domain is responsible for measuring the performance of the SLS portion resided on that domain.

Table 1: Semantic of parameters

Item	Description
Address	The tuple represents the monitoring location
SrcIP	IP of Ingress router to be monitored
SrcPort	Port of Ingress router to be monitored
DesIP	IP of Egress router to be monitored
DesPort	Port of Egress router to be monitored
Class	Class of the service
Gold	Gold service, similar as EF traffic class in DiffServ
Silver	Silver service, similar as AF traffic class in DiffServ
Bronze	Bronze service, similar as BE traffic class in DiffServ
MonTime	The tuple represents monitoring task's schedule
StartTime	Start time of the monitoring task
EndTime	End time of the monitoring task
Interval	Interval of collecting monitoring data
Param	The tuple represents monitoring task detail
Throughput	Throughput between Ingress router and Egress router
Delay	Packet delay between Ingress router and Egress router
Jitter	Packet jitter between Ingress router and Egress router
LossRate	Packet loss rate between Ingress router and Egress router

3.4 SLS Monitor

SLS Monitor in each domain monitors the single domain SLS admitted by that domain. Fig. 3 depicted the detail structure of SLS Monitor. Network Element Monitor employs the measurement approaches built on routers to surveil the network elements in the domain, such as Netflow and SAA provided by Cisco routers. Network Element Monitor stores the data in Monitoring Database. SLS Analyzer, comprising SLS Data Collector, SLS Data Aggregator and SLS Performance Checker, relies on these statistics to assess the service performance and to determine if the SLS is violated. Since the edge router pair is specified in the single domain SLS, SLS Monitor depends on Routing Analyzer to find out the intra-domain edge-to-edge path for the managed service. In practical network, the route between domains is relatively stable, thus we assume that in SLS's period of validity, the single domain SLS traverse the same edge router pairs. For example, single domain SLS SLS_i situated in domain i has ingress edge router $srcER_i$ and egress edge router $desER_i$ for domain i. Routing Analyzer finds out the path for SLS_i in domain i , say $path_i = (srcER_i, R_{i1}, R_{i2}, ..., R_{in}, desER_i)$. The SLS monitoring workflow and the analysis approach are anatomized in our previous work [15].

By SLS Monitor, the approximate causative domain can be localized rapidly. When an alarm is reported to the Alarm Collector indicating a failure happened to the managed service, the monitoring statistics of its related single domain SLSs provides the information about the probable root cause location. If the delay or loss rate between two edge routers in domain i is abnormally high, and other SLS monitors reports normal status, then the root cause occur in domain i with a high probability. In our implementation, the Policy Server provides policies for the delay bound and the loss rate bound in several network load conditions. For instance, in the medium load condition, the delay bound and the loss rate bound are $delay_m$ and $loss_m$, the monitored delay and loss rate in domain i is d_i and l_i . If a SLS traverse domain $i = 1...m$ and reports anomalies, the delay and loss rate in domain $i \in [1,m]$, $d_i - delay_m > \varepsilon_1$ or $l_i - loss_m > \varepsilon_2$, and the delay and loss rate of other domains are within the constraint, then the probability of root cause in domain i is very high.

Network Element Monitor may also report anomalies for SLS Monitor such as link failure, node failure or link utilization threshold is reached. In this case, SLS Monitor analyzes which services are affected by the failure and tries to take countermeasures to prevent the services' performance from deteriorating. For instance, demand Network Management System to initiate a traffic engineering process to change the routing of the service traffic.

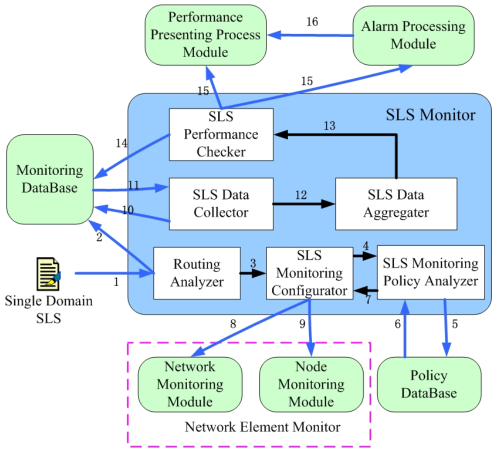


Fig. 3. SLS Monitor

3.5 Alarm Collector

Alarm Collector (AC) is analogous to a complaint department. Service failure alarms are reported to AC indicating faults or congestion happened. Event correlation [5] is used to compress the number of alarms. Then the typical alarms are sent to Fault Diagnostic Toolkit and initiate a fault diagnosis process. In our prototype, alarms are classified into Performance Alarm (PA) and Network Element Alarm (NEA). Performance Alarms are relevant to service performance, such as server is not reachable, server access is too slow, images of the movie cannot be recognized and etc. Network Element Alarms are similar to the alarms of traditional network management system indicating a certain network element failure is observed. Since NEAs have a great impact on service performance, they will be forwarded to SLS Monitor to determine which services are influenced and to take countermeasures, as is called Impact Analysis in [5].

3.6 Fault Diagnostic Toolkit

Fault Diagnostic Toolkit (FDT) starts a fault diagnosis process when alarms are reported. As mentioned in section 2, service unavailability or performance degradation has several probable causes, from server side, client side or the network. Therefore, the fault diagnosis in FDT consists of the following steps:

Server Side Analysis: Obtain the status of server or server farm from the Agent on the server side. If the server side is in a normal condition, go to client side analysis. Otherwise, inform the SP to cope with the service failure.

Client Side Analysis: Get the service status information from the Agent on the client side. If user's computer turns out to be highly occupied or other errors happen on the client side, the user should bear the responsibility. In this case, FDT will inform the user about the faults and guide him to retrieve services of good performance. If nothing is abnormal on client side, go to network analysis.

Network Analysis: Analysis of this stage means that the root cause may happen in the network. FDT retrieves the Global SLS corresponding to the alarm, and delivers it to SLS Monitor. Cooperating with SLS Monitors in other domains along the end-to-end path, SLS Monitor retrieve monitoring data of each single domain SLS. FDT relies on these statistics to determine the approximate root cause position. If an abnormally high delay or packet loss rate is observed in the monitoring statistics of a certain single domain SLS, say SLS_i, the root cause occur in domain i with a high probability. Thus a fault analysis process is initiated in domain i. The network elements along the intra-domain path of SLS_i in domain i, say $path_i = (srcER_i, R_{i1}, R_{i2}, ..., R_{in}, desER_i)$, are likely to be out of order. Sometimes service failure is not caused by these network elements, but by fault propagation from other network portions. Current fault localization techniques can be employed to cope with this case. Thanks to SLS Monitor, the causative region is narrowed down quickly and the accuracy of fault localization is increased.

Though in the initial phase FDT retrieves the Global SLS and the status of each domain in the e2e path, it consumes a little resource because the action only relates to information retrieval, and the fault localization algorithm, the most time-consuming part is not performed. Fault localization is started only after the causative domain in narrowed down. Moreover, each domain is monitored separately and the fault localization is done in a few domains instead of the whole network, thus MDFM is scalable in large network.

4 Implementation

A prototype of MDFM is implemented in a National 863 project of China. As a vital part, MDFM is responsible for localizing the root cause and taking countermeasures rapidly when service failures or performance degradation happens, as a result the user-perceived service down-time is reduced and the service quality is guaranteed. The tesbed is presented in Fig. 4, consisting of five domains with heterogeneous QoS mechanisms and network devices of different vendors. More than

20 routers are deployed in the testbed, and some of them are omitted in Fig. 4 to improve visibility. Fault Analyzer is deployed in each domain.

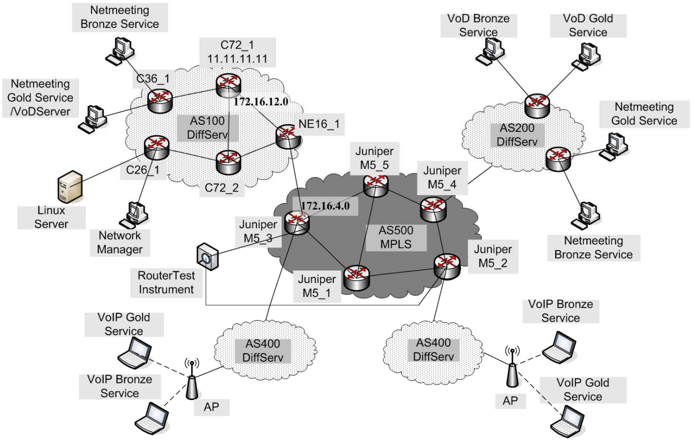


Fig. 4. Testbed

When Video Conference (Netmeeting) service is ongoing, interface 172.16.12.0 is manually disabled and enabled several times to simulate a link malfunction. To produce a congestion situation and cause performance degradation, we use RouterTest instrument manufactured by Agilent and Iperf [16], an open source bandwidth management tool as traffic generators. Iperf injects packets in router 11.11.11.11 and congest link/interface 172.16.12.0. And Routertest generates 256kb UDP packets in the rate of 171.24Mb/s, flooding link 172.16.4.0.

Fig. 5 presents Fault Analyzer’s front-end for network administrators to browse the alarms generated by service failures. Detail content of a Performance Alarm is given in Fig. 6, including the alarm type (PA/NEA), subtype (bandwidth_rate/ drop_rate/link_down/…), status (active/cleared/unknown), level (critical/major/minor /warning/ Information) and etc. Bandwidth Usage, CPU usage and packet loss rate of interface 172.16.4.0 in congestion situation is presented in Fig. 7. The administrator doesn’t need to handle the alarms manually. Instead, FDT performs fault diagnosis automatically.

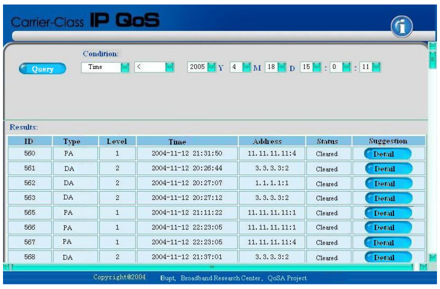
Fig. 8 depicts the performance of Netmeeting service in Gold and Bronze aggregate when congestion happens. Fault management is performed for Gold service but not for Bronze service. From top to bottom, the five diagrams illustrate delay, jitter, packet loss rate, goodput, and network element load. Before the background traffic is generated, their performances are almost the same. But after the traffic is injected to the network, performance alarm of Gold service is reported to MDFM. Based on the SLS monitoring statistics, Fault Analyzer narrows down the causative regions to $AS100$ and $AS500$, and finds out that the root cause is link congestion. Then Fault Analyzer takes immediate countermeasures, demanding the network management system to redirect the service traffic to non-congested links. We can see from the curves that the performance of Gold Service is much better than that of Bronze Service. In addition, the artificial interface malfunction will cause fluctuations in the curve of Bronze service (especially in packet loss rate), but the curve of Gold service is comparatively smooth.

Experiments are also conducted for other services such as VoIP and VOD. The results are omitted here due to space constraint. On the whole, our fault analyzer can locate the root cause and take countermeasures quickly, as a result prevents service performance degradation when service failure occurs.

5 Conclusion

Due to the characteristics of Internet service, service fault management has new requirements. We analyze the challenges and propose a multi-domain service fault management framework MDFM in this paper. By SLS decomposition and monitoring based on network capability, root cause analysis of the service failure is focused rapidly on a certain domain, solving the problem of uncertain fault propagation probabilities among domains. In addition, our framework takes both the end systems and network failures into account. A prototype of MDFM is implemented and proves the feasibility and efficiency of our fault management framework.

Security is a necessity for commercial management system. In our future work, security mechanisms will be added to MDFM such as encryption, digital signature and access control to prevent illegal access of the Fault Analyzer. And Agents' abilities will be extended to find out the end-systems' failures.

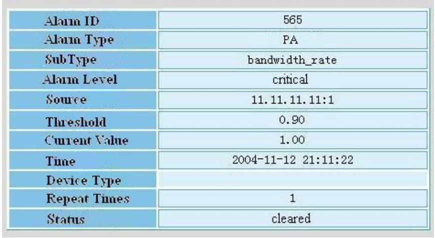


Condition: Time < 2005 Y 4 M 18 D 15 : 0 : 11

Results:

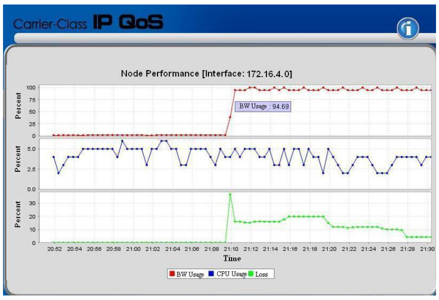
ID	Type	Level	Time	Address	Status	Suggestion
560	PA	1	2004-11-12 21:31:50	11.11.11.11:4	Cleared	Detail
561	DA	2	2004-11-12 20:26:44	3.3.3.3:2	Cleared	Detail
562	DA	2	2004-11-12 20:27:07	1.1.1.1:1	Cleared	Detail
563	DA	2	2004-11-12 20:27:12	3.3.3.3:2	Cleared	Detail
565	PA	1	2004-11-12 21:11:22	11.11.11.11:1	Cleared	Detail
566	PA	1	2004-11-12 22:23:05	11.11.11.11:1	Cleared	Detail
567	PA	1	2004-11-12 22:23:05	11.11.11.11:4	Cleared	Detail
568	DA	2	2004-11-12 21:37:01	3.3.3.3:2	Cleared	Detail

Copyright@2004 Bupt, Broadband Research Center, QoSA Project



Alarm ID	565
Alarm Type	PA
SubType	bandwidth_rate
Alarm Level	critical
Source	11.11.11.11:1
Threshold	0.90
Current Value	1.00
Time	2004-11-12 21:11:22
Device Type	
Repeat Times	1
Status	cleared

Fig. 5. Fault Analyzer (Alarms Browser) **Fig. 6.** A Performance Alarm Example



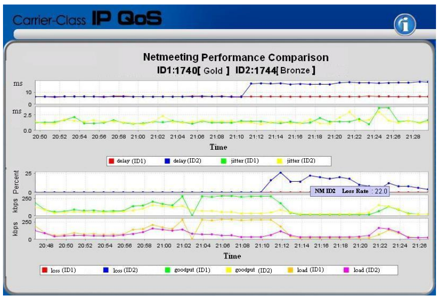


Fig. 7. Interface Usage in Congestion **Fig. 8.** Netmeeting SLS Monitoring Result

References

1. Yemini, S.A., Kliger, S., Mozes, E., Yemini, Y., Ohsie, D.: High speed and robust event correlation, Communications Magazine, IEEE Volume 34, Issue 5, May 1996 Page(s):82 - 90
2. M. Steinder and A. S. Sethi: The present and future of event correlation: A need for end-to-end service fault localization, World Multi-Conf. Systemics, Cybernetics, and Informatics (SCI), Orlando, FL, 2001
3. M. Steinder and A. S. Sethi: Probabilistic Fault Localization in Communication Systems Using Belief Networks, IEEE/ACM Transactions on Networking, Vol.12, No. 5, October 2004
4. M. Steinder and A.S. Sethi: Multi-Domain Diagnosis of End-to-End Service Failures in Hierarchically Routed Networks, Lecture Notes in Computer Science Vol. LNCS-3042, (2004), pp. 1036-1046, Heidelberg: Springer-Verlag.
5. Andreas Hanemann, Martin Sailer, David Schmitz: Assured Service Quality by Improved Fault Management - Service-Oriented Event Correlation, Proceedings of the 2nd international conference on Service oriented computing, November 2004
6. Qinzheng Kong, Chen, G., Hussain, R.Y.: A Management Framework for Internet Services, Network Operations and Management Symposium, 1998. NOMS 98., IEEE , Volume: 1 , 15-20 Feb. 1998 Pages:21 - 30 vol.1
7. Darst, C., Ramanathan, S.: Measurement and Management of Internet Services, Integrated Network Management, 1999. Proceedings of the Sixth IFIP/IEEE International Symposium on Distributed Management for the Networked Millennium, 24-28 May 1999, Pages:125 - 140
8. Deborah Caswell, Srinivas Ramanathan: Using Service Models for Management of Internet Services, IEEE Journal on Selected Areas in Communications, Volume: 18 , Issue: 5, Pages:686 – 701, May 2000
9. Alexandre Bronstein, Joydip Das, etc, Self-Aware Services: Using Bayesian Networks for Detecting Anomalies in Internet-based Services, Technical Report HPL-2001-23 (R.1), HP Laboratories Palo Alto, “www.hpl.hp.com/techreports/2001/HPL-2001- 23R1.ps”, 2001
10. IBM Redbook, Business Service Management Best Practices, http://IBM.com/redbooks
11. Rainer Hauck, Igor Radisic: Service Oriented Application Management-Do Current Techniques Meet The Requirements?, New Developments in Distributed Applications and Interoperable Systems: 3rd IFIP International Working Conference (DAIS2001)
12. K. Nichols and B. Carpenter: Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification, RFC 3086, April 2001
13. Xiaohui Huang, Yu Lin, Wendong Wang, Xirong Que, Shiduan Cheng, Li Jiao, Yidong Cui: QoSjava: An Open and Scalable Architecture Decoupling QoS Requirements from QoS Techniques, draft-bupt-qosjava-arch-02.txt, http://www.ietf.org/internet-drafts/ draft-bupt-qosjava-arch-02.txt
14. Xiaohui Huang, Yu Lin, Wendong Wang, Shiduan Cheng: PDB-Based SLS Decomposition in Heterogeneous IP Network, Proceedings of 2004 IEEE International Workshop on IP Operations & Management.
15. Junfeng Xiao, Yidong Cui, Wendong Wang, Shiduan Cheng: A Service Level Specification (SLS) Monitoring System in Multiple Services IP Network, High technology Letters, ISSN 1002-0470, published by Executive Office of the Journal, Institute of Scientific and Technical Information of China, to appear.
16. Iperf, University of Illinois, “http://dast.nlanr.net/Projects/Iperf/”