

УДК 621.397

УПРАВЛЕНИЕ СЕРВИСАМИ СРЕДСТВАМИ SNMP И RMONv2



[Е.В. ДУРАВКИН](#)

Харьковский национальный
университет радиоэлектроники

Запропонована архітектура системи управління сервісами на базі використання протоколу SNMP. Система, що пропонується, дозволяє виконувати розподіл заявок на надання сервісів з урахуванням завантаження серверів, що реалізують екземпляри сервісів.

Architecture for service management system based on SNMP is proposed. The system allows to distribute requests of services, taking into account the workload of servers.

Предложена архитектура системы управления сервісами на базе использования протокола SNMP. Предлагаемая система позволяет выполнять распределение заявок на предоставление сервисов с учетом загрузки серверов, реализующих экземпляры сервисов.

Введение

На сегодняшний день при разработке мультисервисных сетей значительное внимание уделяется вопросам унификации и обеспечения качества обслуживания. Для решения проблемы унификации в первую очередь используются специальные подходы при проектировании и организации взаимодействия между компонентами в системе. На сегодняшний день одним из таких подходов, получивших наибольшее распространение, является сервис-ориентированная архитектура (SOA) [1].

I. Постановка проблемы и анализ публикаций

В соответствии с SOA выполнено большое количество как корпоративных, так и региональных мультисервисных сетей. Однако, наряду с большим количеством достоинств, у данного подхода есть и некоторые недостатки. В качестве одного из основных недостатков можно выделить отсутствие механизмов обеспечения требуемого уровня QoS (Quality of Service), надежности, доступности и отказоустойчивости сервисов в сети [5,6].

В то же время на рынке уже достаточно долго существуют системы сетевого управления (так называемые решения OSS/BSS – Operation Support System/Business Support System), основной задачей которых является обеспечение деятельности диспетчера сети в контексте учёта и планирования ресурсов сети, управление услугами, управление QoS (IBM Tivoli, HP OpenView). Однако существующие OSS/BSS решения в основном позволяют лишь собирать статистику состояний узлов сети без какой либо ее обработки и выработки управляющих воздействий в соответствии с предоставляемыми услугами и требуемым уровнем обслуживания.

Таким образом, для обеспечения заданных дисциплин QoS и уровней доступности и надежности предоставляемых сервисов необходимо усовершенствовать су-

существующие системы сетевого управления в области обработки информации о состоянии элементов мультисервисной SOA сети и выработки управляющих воздействий.

Как было отмечено выше, для обеспечения требуемого уровня QoS необходимо усовершенствовать алгоритмы работы существующих систем сетевого управления. В настоящей работе предлагается алгоритм сбора и обработки информации о состоянии узлов мультисервисной сети и последующей выработки команд для корректировки параметров ESB (Enterprise Service Bus) и распределения потока заявок.

Предлагаемый алгоритм основан на работе системы сетевого управления, использующей протокол SNMPv2 (Simple network management protocol) в качестве протокола обмена сообщениями и передачи команд и агентов мониторинга, поддерживающих стандарт RMONv2 (Remote Network MONitoring).

II. Архитектура системы управления сервисами

На сегодняшний день протокол SNMP получил наибольшую популярность в качестве основы для систем сетевого управления и RMON в качестве базы данных хранения параметров объектов управления. RMON предоставляет набор статистических данных и функций для обмена данными.

В современных системах сетевого управления используются 2 версии RMON [3, 4]. Отличие RMONv2 от RMONv1 заключается в том, что RMONv2 позволяет собирать информацию о верхних уровнях OSI (с сетевого до прикладного). В частности, в состав базы RMONv2 входят такие параметры как *Network layer matrix*, *Application layer host*, *Application layer matrix*, *User history*. Анализ данных параметров позволяет получать не только информацию о состоянии сетевых интерфейсов, но и собирать статистику работы протоколов прикладного уровня. Следовательно, RMONv2 можно применять для мониторинга и управления сервисами в мультисервисных сетях, использующих технологию SOA. На рис. 1 приведена структурная схема системы управления сервисами в мультисервисной сети. Основой данной системы являются frontend и прокси серверы, выполняющие обработку входящих запросов и их распределение по серверам-экземплярам сервисов.

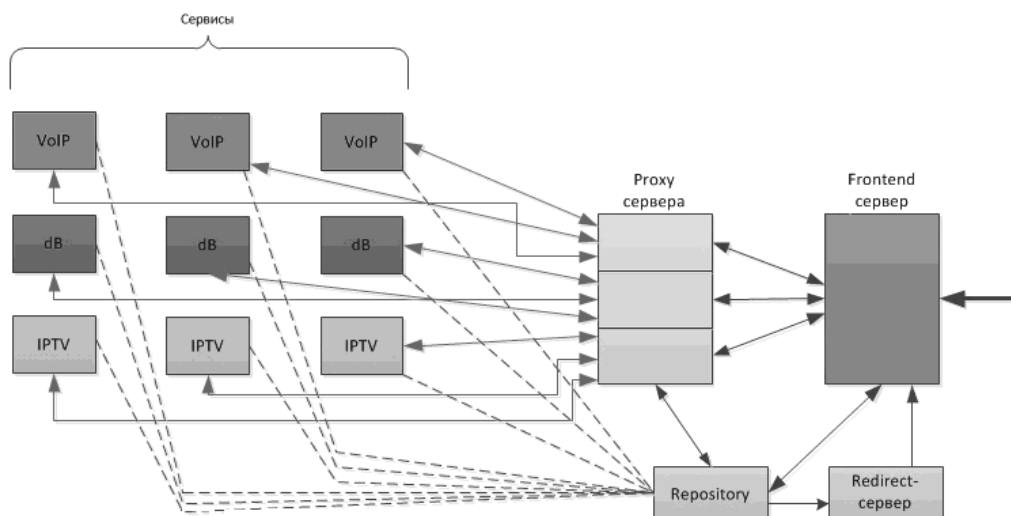


Рис. 1. Структурная схема системы управления сервисами

Эффективность работы (время реакции, время отклика) данной системы во многом зависит от алгоритмов выбора экземпляра сервиса для обслуживания заявки. В простейших случаях в качестве таких алгоритмов могут выступать случайный выбор или кольцевой алгоритм загрузки. Однако эти алгоритмы не учитывают состояния сервиса, а следовательно, могут приводить к перегрузке одних узлов и недостаточной загрузке других узлов системы.

Для устранения данного недостатка предлагается алгоритм распределения заявок пользователей на основе анализа состояния (загруженности) узлов системы управления сервисами.

Реализация предлагаемого алгоритма предполагает включение в состав системы управления сервисами агентов мониторинга состояния серверов и управляющего менеджера, работающих на основе протокола SNMP. Задачей агентов мониторинга является сбор информации о состоянии узлов из базы RMONv2, а задачей менеджера является обработка информации о состоянии узлов и принятие решения о распределении потока заявок по узлам (экземплярам сервисов). Исходя из решаемых задач, агенты мониторинга представляют собой программные модули, реализованные на стороне узлов, а менеджер – часть системы управления сервисами.

Одним из важных моментов при организации такой системы мониторинга и управления является выбор интервалов опроса агентов (T_o). Важность данного параметра обусловлена тем, что некорректный его выбор может породить значительное количество служебного трафика с системе в ущерб основному, критически чувствительному к задержкам.

Обычно объем служебного трафика не должен превышать 10% от общей пропускной способности канала связи [7]. Следовательно, зная размер пакета SNMP (200 байт) и количество агентов мониторинга (N), можно вычислить объем служебного трафика:

$$V = \frac{200N}{T_o}.$$

Другой подход организации работы системы сетевого управления заключается в том, что с целью снижения сетевой нагрузки получение информации о состоянии узлов системы управления сервисами предлагается выполнять не посредством опроса агентов мониторинга, а настройкой соответствующих пороговых значений в MIB. При превышении порогового значения агент мониторинга самостоятельно будет формировать сообщение менеджеру. Пороги устанавливаются с помощью явного задания значений базовых показателей или путем применения аналитических методов.

Одним из наиболее эффективных методов настройки пороговых значений является метод последовательной оптимизации. Суть метода заключается в том, что на первом этапе порог срабатывания сигнала *Alarm* по заданному параметру устанавливается низким, и последовательно повышается, пока его значение не станет приемлемым для системы.

Необходимо заметить, что значение порога зависит от специфики работы узла (типа реализуемого сервиса), на котором выполняется мониторинг. Для анализа чув-

ствительности сервисов различного рода к значениям порогов оповещения разработана имитационная модель системы управления сервисами (рис. 1). На вход системы подавался нарастающий поток заявок и анализировался процент отброшенных заявок в зависимости значения порога (параметр *etherStatsDropEvents* группы *Statistics*) для серверов, предоставляющих услуги доступа к базам данных (БД) и VoIP. Как видно из графика, приведенного на рис. 2, при одинаковой динамике зависимости потерь заявок для мультимедийных сервисов (VoIP) и сервисов передачи данных (БД) значения потерь для мультимедийных сервисов выше, что обусловливается особенностями организации транспортного уровня данных сервисов. Следовательно, данный факт нужно учитывать при настройках агентов мониторинга на различных сервисах.

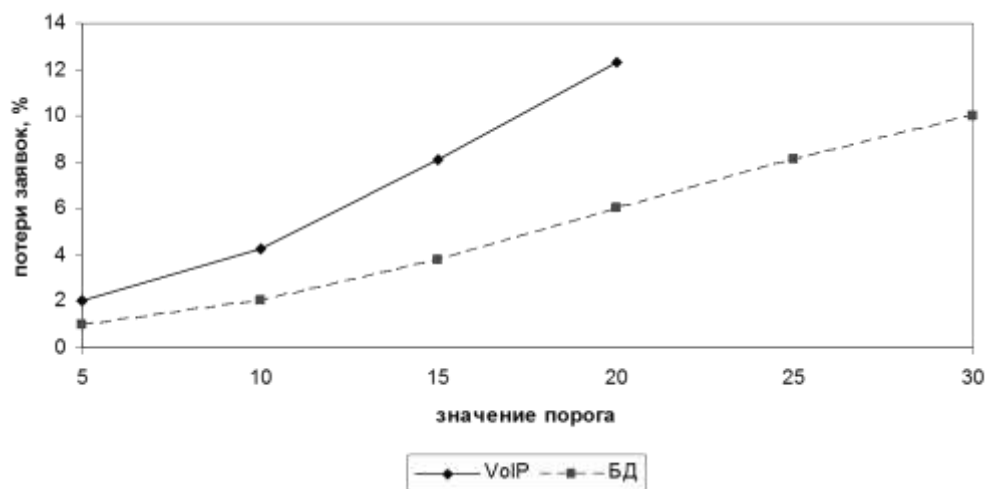


Рис. 2. Зависимость потерь заявок на обслуживания от значения порогов предупреждения

Для обеспечения показателей отказоустойчивости и надежности предоставления сервисов система управления должна обладать возможностями обнаружения логических сбоев в работе сервисов и их восстановления в случае отказов.

Обнаружение логических сбоев в работе сервиса можно выполнять на основе анализа параметров *alHost*, *alMatrix*, *usrHistory*. Указанные параметры позволяют отслеживать количество сбоев на прикладном уровне для указанного протокола. Следовательно, превышение порога по одному из этих параметров приводит к появлению сигнала *Alarm* и выработке менеджером решения о необходимости останова сервиса и его восстановления.

Повышение доступности сервисов достигается увеличением числа его копий в сегментах сети, где генерируется наибольшее количество запросов к сервису. Параметры RMONv2 *nlHost* и *nlMatrix* позволяют собирать статистику по количеству обращений к узлу с других узлов и подсетей. Следовательно, менеджер системы управления сервисами может определить, что в какой-то момент времени к сервису возрос поток заявок из другой подсети. В случае, если в удаленной подсети доступны ресурсы для системы сетевого управления, то она выполняет репликацию сервиса. Выполнение репликации позволит не только повысить доступность сервиса, но и снизить объемы трафика между подсетями.

Общая архитектура предлагаемой системы управления сервисами приведена на рис. 3.

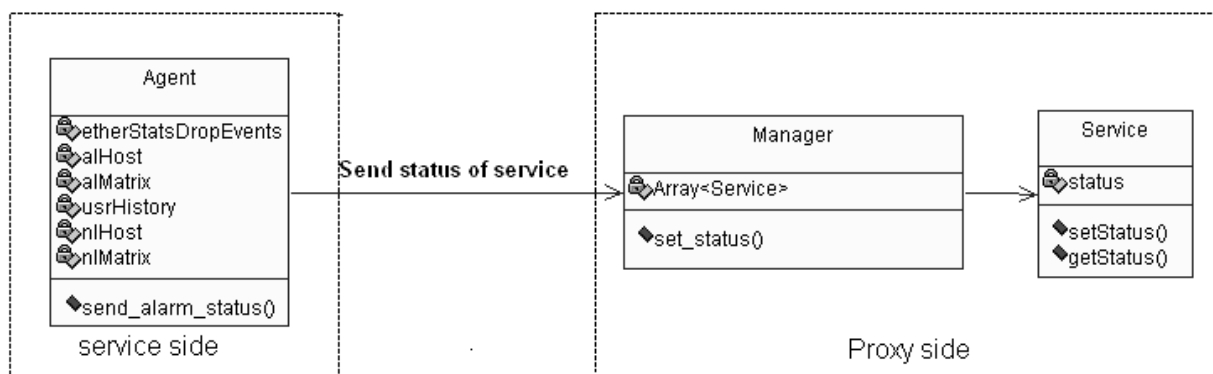


Рис. 3. Общая архитектура системы управления сервисами

Система управления сервисами в соответствии с идеологией протокола SNMP состоит из двух основных классов – «Агент мониторинга» и «Менеджер агентов».

Объект класса «Агент мониторинга» отслеживает состояние объектов базы RMON и при превышении пороговых значений отправляет Менеджеру сообщение об изменении состояния экземпляра сервиса.

Объект класса «Менеджер агентов» в качестве одного из атрибутов содержит массив, описывающий состояние экземпляров сервисов. Обновление содержимого данного массива происходит при получении сообщений от агентов мониторинга.

Диаграмма последовательностей обновления информации о статусе экземпляра сервиса приведена на рис. 4.

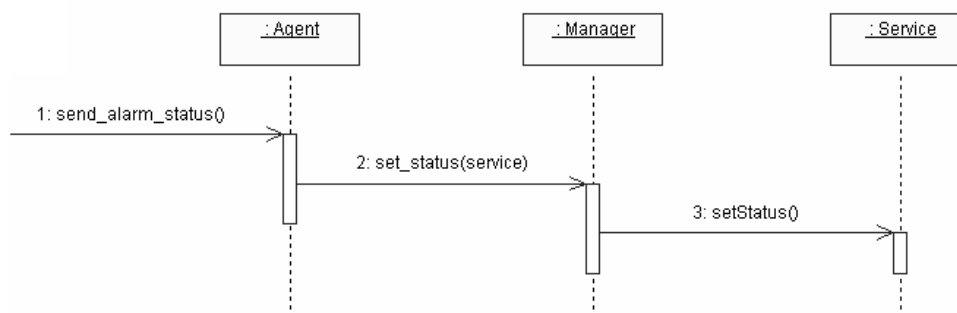


Рис. 4. Диаграмма последовательностей обновления статуса экземпляра сервиса

При получении запроса на предоставление сервиса, прокси-сервер анализирует состояние экземпляров сервисов и выполняет перенаправление запроса за экземпляр с наилучшими характеристиками. Диаграмма последовательностей выбора экземпляра сервиса для обработки запроса приведена на рис. 5.

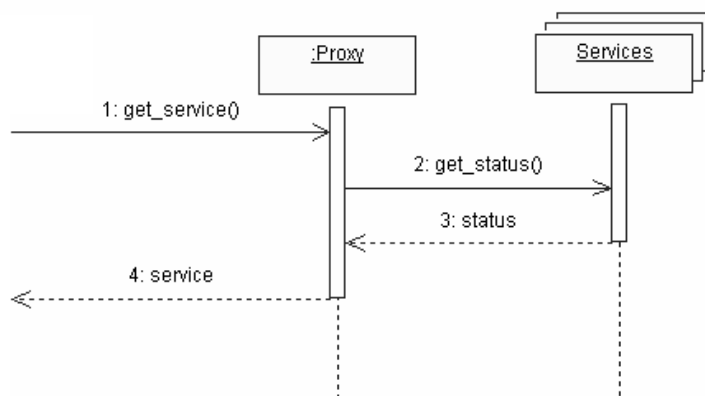


Рис. 5. Диаграмма выбора экземпляра сервиса

Выводы

В статье предложена архитектура системы управления сервисами, которая в отличие от известных не только выполняет мониторинг состояния на нижних уровнях OSI, но позволяет распределять запросы на обслуживание заявок в соответствии с текущим состоянием экземпляров сервисов. Состояние экземпляра сервиса проверяется агентом мониторинга, который реализуется на стороне экземпляра сервиса. Агент мониторинга отслеживает значение параметров etherStatsDropEvents, allHost, allMatrix, usrHistory, nlHost, nlMatrix базы RMON и при превышении любым из параметров порогового значения отправляет соответствующее сообщение управляющему менеджеру. Данный подход позволяет повысить эффективность работы прокси-сервера распределяющего заявки на обслуживания, не повышая значительно объем служебного трафика.

Список литературы:

1. Александров А. SOA для сетей [Электронный ресурс] // Открытые системы. — 2007. — № 3. — Режим доступа: <http://www.osp.ru/os/2007/03/4177497/>.
2. Кузнецов С. SOA с гарантией качества [Электронный ресурс] // Открытые системы. — 2008. — № 7. — Режим доступа: <http://www.osp.ru/os/2008/07/5479587/>.
3. Waldbusser S. Remote Network Monitoring Management Information Base. — RFC 2819, 2000.
4. Waldbusser S. Remote Network Monitoring Management Information Base Version 2 using SMIv2. — RFC 2021, 1997.
5. Zhu K., Duan Z., Wang J. Quality of service in web services discovery // IEEE Symposium on Advanced Management of Information for Globalized Enterprises, 2008. — P. 1–5.
6. Web services quality factors v1.0 [Электронный ресурс] // Advance Open Standard for the information society. — Режим доступа: [http://www.oasis-open.org/committees/download.php/38611/WS-Quality Factors v1.0 cd02.zip](http://www.oasis-open.org/committees/download.php/38611/WS-Quality%20Factors%20v1.0%20cd02.zip).
7. Бломмерс Д. OpenView. Network Node Manager. Разработка и реализация корпоративного решения. — Интернет-университет информационных технологий – ИНТУИТ.ру, 2005. — 264 с.