

Используем систему мониторинга сети OpenNMS

Андрей Семенов

Что ж, вопросы с установкой решены (см. №5 за 2008 г.), и система openNMS работает. Как же теперь сделать ее более функциональной, а работу с ней более удобной? Запасаемся временем и терпением, не забываем подключить к работе голову и руки – и вперед!

Возможности веб-интерфейса

Какую же информацию мы можем получить после начальной настройки OpenNMS через веб-интерфейс? На главной странице веб-интерфейса, на которую мы попадаем сразу после авторизации, представлена информация о текущем состоянии наблюдаемых устройств, и если какое-то устройство, интерфейс или сервис недоступны, то эта информация выделяется цветом (см. **рис. 1**).

В верхней части интерфейса расположено основное навигационное меню, с помощью которого можно просмотреть список всех проверяемых узлов (меню «Node List»), либо с помощью поискового интерфейса (меню «Search») найти конкретное устройство по различным критериям поиска: IP-адресу, наименованию, предоставляемому сервису и др.

Каждому узлу можно сопоставить дополнительную инвентарную информацию и затем просматривать ее через меню «Assets» (имущество). Путем несложной навигации можно узнать подробную статистику о текущих и исправленных отказах (меню «Outages») устройств и сервисов, об авариях (меню «Alarms»), а также обо всех событиях (меню «Events»), зарегистрированных системой.

В меню «Reports» (отчеты) можно просмотреть отчеты о доступности и графики, полученные при опросах устройств. Это могут быть и основанные на ICMP-запросах RoundTrip Ping и StrafePing (статистика по потерянным пакетам), и графики, основанные на данных, собранных с помощью SNMP. Имеется возможность составлять собственные отчеты (Custom Reports), основанные на собранных данных.

В меню «Surveillance» (слежение) можно сгруппировать информацию об узлах сети по определенным признакам в виде таблицы. Можно, например, категоризировать устройства по территориальной принадлежности и типу оборудования. По умолчанию ни один узел не принадлежит какой-либо категории, добавление узла в категорию можно произвести через администраторский интерфейс (меню «Admin»).

В меню «DashBoard» (панель инструментов) на основе категоризации из меню «Surveillance» собрана дополнительная информация о состоянии устройств, такая, как текущие аварии, состояние об отказах устройства за последние 24 часа и статистические графики.

С помощью меню «Admin» существует возможность изменять многие настройки OpenNMS, но возможностей

веб-интерфейса в некоторых случаях бывает недостаточным, поэтому возникает необходимость прямого редактирования конфигурационных файлов.

Группировка узлов по определенным признакам с помощью таблиц слежения

Визуальное представление многих данных мониторинга, просматриваемых через веб-интерфейс, можно изменять в некоторых пределах. Одна из удобных функций заключается в возможности группировки узлов сети по различным признакам в виде таблицы слежения. В меню «Surveillance» (слежение) уже задано разбиение на категории по умолчанию (см. **рис. 2**), однако сетевые устройства пока не привязаны к существующим категориям слежения.

Добавить устройство в необходимую категорию по определенным признакам можно через меню «Admin → Manage Surveillance Categories» (см. **рис. 3**).

В строках и столбцах таблицы слежения определяют признаки наблюдаемых устройств, а в местах пересечения конкретной строки и столбца отображается количество узлов, удовлетворяющих этим признакам. Можно создать необходимое количество собственных категорий слежения, на основе которых в дальнейшем построить собственную таблицу (таких таблиц может быть несколько). Для создания таблицы слежения или модификации существующей необходимо отредактировать конфигурационный файл `surveillance-views.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<surveillance-view-configuration
  xmlns:this="http://www.opennms.org/xsd/config/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.opennms.org/xsd/config/surveillance-views
    http://www.opennms.org/xsd/config/surveillance-views.xsd"
  default-view="by Priority and Office" >
  <views >
    <view name="default" refresh-seconds="300" >
      <rows>
        <row-def row="1" label="Main Office" >
          <category name="MainOffice" />
        </row-def>
        <row-def row="2" label="Filial 1" >
          <category name="Filial-1" />
        </row-def>
        <row-def row="3" label="Filial 2" >
          <category name="Filial-2" />
        </row-def>
        <row-def row="4" label="Filial 3" >
          <category name="Filial-3" />
        </row-def>
      </rows>
      <columns>
        <column-def col="1" label="Routers" >
          <category name="Routers" />
        </column-def>
        <column-def col="2" label="Servers" >
          <category name="Servers" />
        </column-def>
        <column-def col="3" label="Switches" >
          <category name="Switches" />
        </column-def>
      </columns>
    </view>

    <view name="by Priority and Office" refresh-seconds="300" >
      <rows>
```

OpenNMS Web Console - Mozilla Firefox

http://localhost:8980/opennms/index.jsp

Web Console
User: **admin** (Notices Off) - Log out
18.05.2008 14:44 VLAST

Node List Search Outages Path Outages Dashboard Events Alarms Notifications Assets Reports Charts Surveillance Admin Help

Home

Nodes with Outages
Server2 (3 weeks)

Percentage change over past 24 hours

Categories	Outages	Availability
Network Interfaces	1 of 10	85,814%
Web Servers	0 of 5	96,358%
Email Servers	0 of 4	99,809%
DNS and DHCP Servers	0 of 3	99,940%
Database Servers	0 of 0	100,000%
JMX Servers	0 of 0	100,000%
Other Servers	0 of 4	99,828%
Total	Outages	Availability
Overall Service Availability	1 of 32	93,638%

Notification
You: No outstanding notices (Check)
All: No outstanding notices (Check)
On-Call Schedule

Resource Graphs
-- Choose a node --

KSC Reports
No KSC reports defined

OpenNMS Copyright © 2002-2008 The OpenNMS Group, Inc. OpenNMS® is a registered trademark of The OpenNMS Group, Inc.

Рисунок 1. Главная страница веб-интерфейса

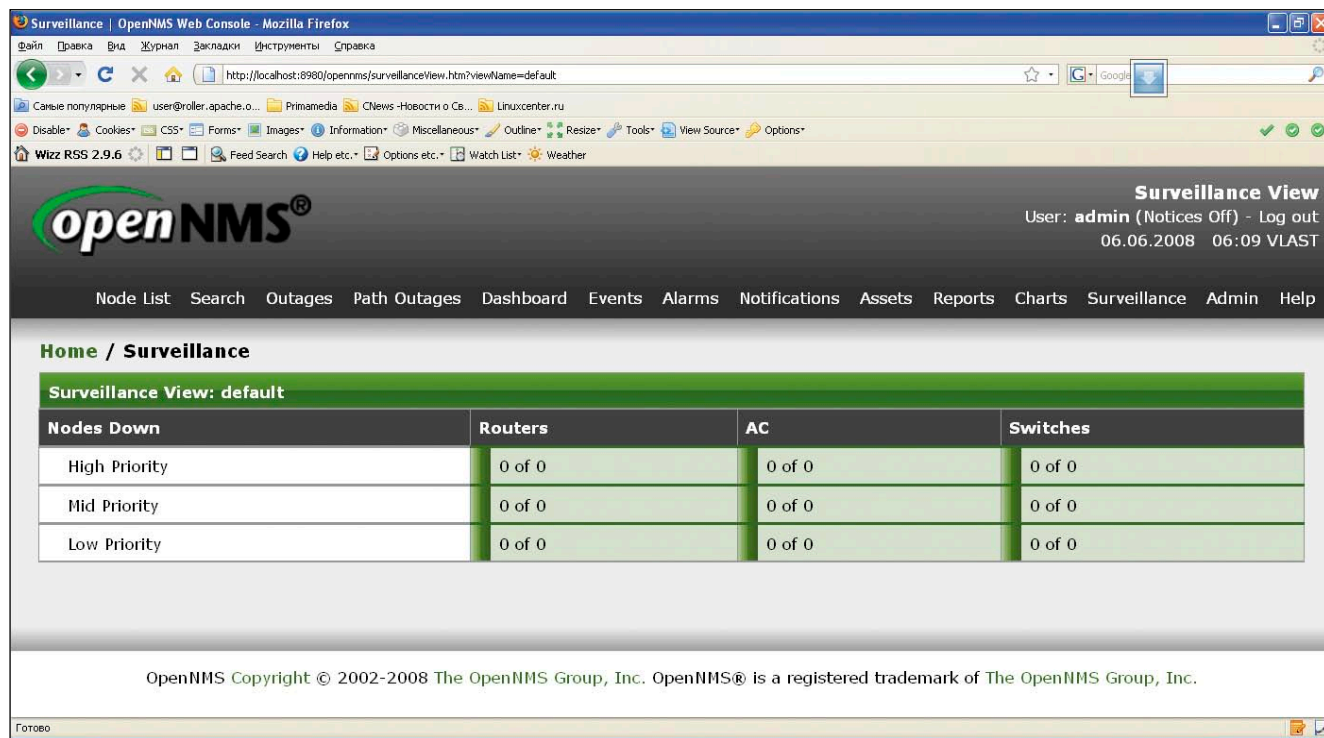


Рисунок 2. Вид меню «Surveillance» после установки системы

```
<row-def row="1" label="Main Office" >
<category name="MainOffice" />
</row-def>
<row-def row="2" label="Filial 1" >
<category name="Filial-1" />
</row-def>
<row-def row="3" label="Filial 2" >
<category name="Filial-2" />
</row-def>
<row-def row="3" label="Filial 3" >
<category name="Filial-3" />
</row-def>
</rows>
<columns>
<column-def col="1" label="High Priority" >
<category name="HighPriority" />
</column-def>
<column-def col="2" label="Mid Priority" >
<category name="MidPriority" />
</column-def>
<column-def col="3" label="Low Priority" >
<category name="LowPriority" />
</column-def>
</columns>
</view>
</views>
</surveillance-view-configuration>
```

В тег `<views>` вложен тег `<view>`, в котором описываются строки и столбцы таблицы категорий. В параметре `name` тега `<view>` задается название таблицы категорий. В теге `<rows>` описываются строки таблицы с помощью вложенных тегов `<row-def>`, а в теге `<columns>` с помощью `<column-def>` описываются столбцы. В тегах `<row-def>` и `<column-def>`, которые являются строками и столбцами таблицы категорий, описываются категории узлов с помощью тегов `<category>`. Данные в тегах `<category>` должны соответствовать данным, заполненным через меню «Admin → Manage Surveillance Categories» (данные хранятся в таблице `categories` базы данных PostgreSQL), иначе при попытке перейти в меню «Surveillance» мы получим ошибку.

В приведенном выше примере описаны два разных разбиения на категории – «default» и «by Priority and Office»

(см. рис. 4), однако вы можете создать таблицы слежения по вашим потребностям.

В корневом теге `<surveillance-view-configuration>` в параметре `default-view` можно задать отображаемую по умолчанию таблицу слежения (в приведенном выше конфигурационном файле этому параметру присвоено имя таблицы «by Priority and Office», которая будет отображаться при переходе в меню «Surveillance» и «DashBoard» (см. рис. 5).

Настройка уведомлений

В openNMS существует возможность использовать систему уведомлений, чтобы оповещать пользователей о происходящих событиях. Основным методом уведомлений является отправка e-mail-сообщений пользователю, но существует и ряд других методов, например, отправка POST/GET-запросов на веб-сервер, отправка уведомлений по протоколу XMPP (jabber), пересылка уведомлений посредством запуска внешней программы (подобным образом можно отправить SMS-сообщение с помощью GSM-модема) и уведомления с помощью формирования SNMP traps.

Мы рассмотрим классический способ с отправкой уведомлений на электронную почту, для чего нам понадобится внести необходимые настройки в следующие конфигурационные файлы:

- **javamail-configuration.properties** – в файле содержатся настройки почтового сервера и учетной записи (почтового ящика);
- **destinationPaths.xml** – в файле описывается, кто получает уведомления;
- **notifd-configuration.xml** – здесь описываются глобальные настройки службы уведомлений;
- **notificationCommands.xml** – в файле описываются методы уведомления, такие, как e-mail-оповещения, XMPP и др.;

■ **notification.xml** – здесь содержится описание уведомлений.

Будем использовать тип доставки javaEmail (уведомления на электронную почту) для оповещения о происходящих в системе событиях. Для этого изменим настройки файла `javamail-configuration.properties`, чтобы отправлять сообщения через ящик на сервере Google Mail, например (однако это может быть и ваш корпоративный почтовый сервер):

```
org.opennms.core.utsils.useJMTA=false
org.opennms.core.utsils.transport=smtps
org.opennms.core.utsils.mailHost=smtp.gmail.com
org.opennms.core.utsils.smtpport=465
org.opennms.core.utsils.authenticate=true
org.opennms.core.utsils.authenticateUser=user@gmail.com
org.opennms.core.utsils.authenticatePassword=userpass
org.opennms.core.utsils.starttls.enable=true
org.opennms.core.utsils.smtpssl.enable=true
org.opennms.core.utsils.messageContentType=text/html
org.opennms.core.utsils.charset=windows-1251
```

Google Mail использует SMTP over SSL (TLSv1 и SSLv3), поэтому параметр `org.opennms.core.utsils.smtpssl.enable` выставлен в `true`, в качестве транспорта (`org.opennms.core.utsils.transport`) используется SMTPS, а порт для подключения (`org.opennms.core.utsils.smtpport`) используется стандартный для SMTPS – 465. В более простых случаях, когда используется обычный SMTP-протокол без шифрования, необходимо будет заменить номер порта на 25, транспорт –

на `smtp` и параметры `smtpssl.enable` и `starttls.enable=true` выставить в `false`.

В файле `destinationPaths.xml` содержится информация об адресах доставки уведомлений:

```
<?xml version="1.0" encoding="UTF-8"?>
<destinationPaths xmlns=
"http://xmlns.opennms.org/xsd/destinationPaths">
<ns1:header xmlns:ns1="http://xmlns.opennms.org/xsd/types">
<rev xmlns="">1.2</rev>
<created xmlns="">10 Июнь 2008 г. 4:43:30 GMT</created>
<mstation xmlns="">localhost</mstation>
</ns1:header>
<path name="Email-Admin" initial-delay="0s">
<target interval="0s">
<name xmlns="">Admin</name>
<command xmlns="">javaEmail</command>
</target>
</path>
</destinationPaths>
```

Атрибут `name` тега `<path>` определяет имя блока адреса доставки. Значение данного атрибута также используется в файле `notifications.xml` в качестве адреса доставки при описании конкретного вида уведомлений. Внутри каждого тега `<path>` можно описать несколько вложенных тегов `<target>`, в которых с помощью тега `<name>` можно описать адресатов, которым будут посылаться уведомления, а также способ доставки (в нашем случае `javaEmail`) с помощью тега `<command>`. В качестве адресата может выступать как отдельный пользователь, так и группа или роль. Добавить учетную запись пользователя, группы и роли можно через

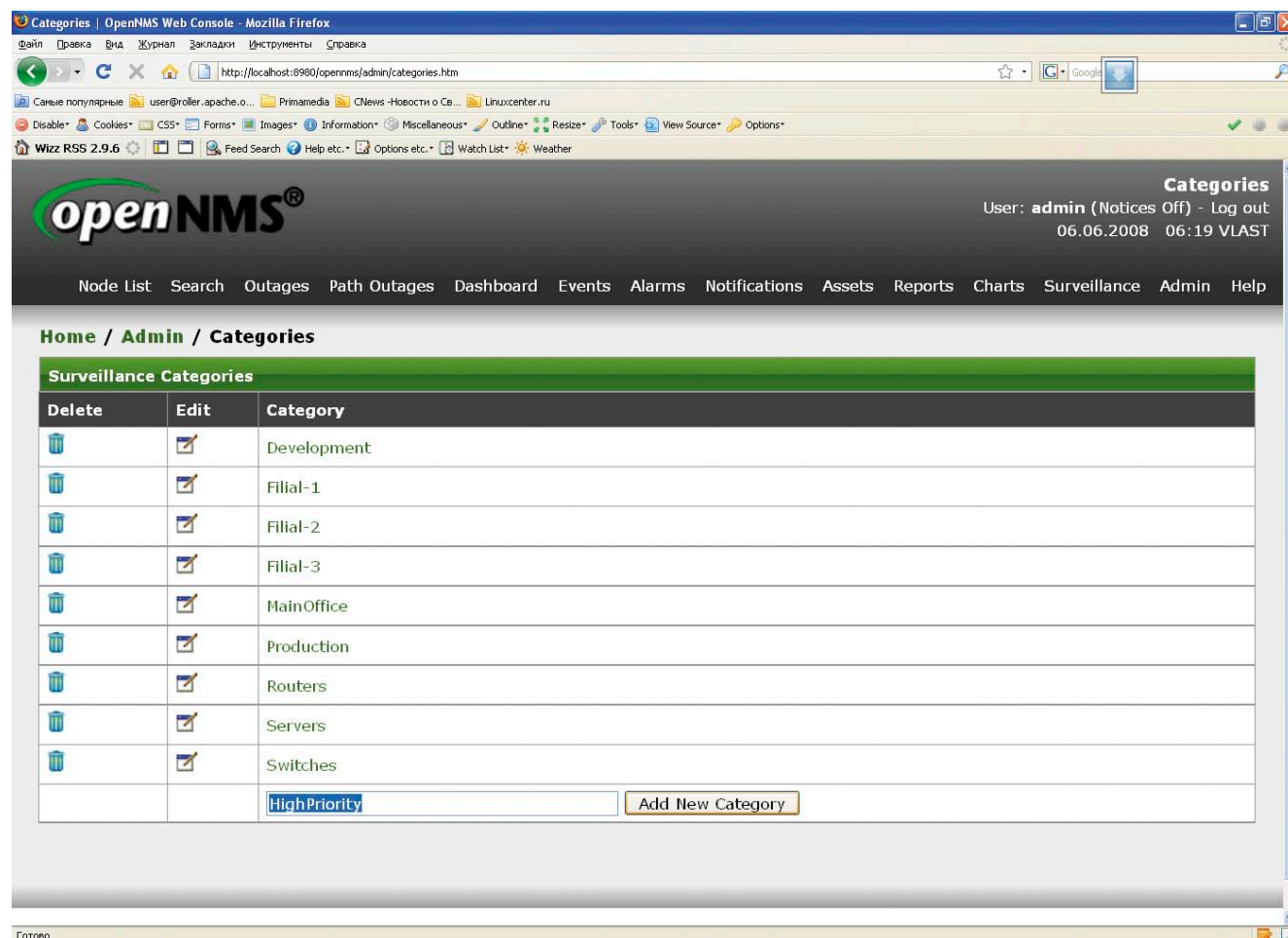


Рисунок 3. Добавление узлов в категории по определенным признакам

веб-интерфейс (меню «Admin → Configure Users, Groups and Roles») или через конфигурационные файлы users.xml (пользователи) и groups.xml (группы и роли).

Для обеспечения более гибкой отправки уведомлений в файле destinationPaths.xml можно описать несколько дополнительных тегов <path>:

```
<path name="Email-Filial1" initial-delay="0s">
  <target interval="0m">
    <name xmlns="">Filial1</name>
    <autoNotify xmlns="">auto</autoNotify>
    <command xmlns="">javaEmail</command>
  </target>
</path>
```

В приведенном выше коде создается путь для отправки уведомлений на адрес абонента Filial1. Конечно, учетная запись пользователя (или группы, роли) Filial1 должна быть предварительно создана, в таком случае уведомления будут отправляться, например, одному или нескольким сотрудникам филиала №1. Эти сотрудники будут получать почтовые сообщения в случае проблем с сетевыми устройствами или сервисами.

Настройки уведомлений задаются в файле notifications.xml. По умолчанию здесь уже заданы стандартные типы уведомлений, которые отправляются на адрес доставки Email-Admin (то есть всем пользователям группы Admin). Теперь добавим собственное уведомление, которое будет отправляться по адресу доставки Email-Filial1:

```
<notification name="Filial1-nodeDown" status="on" ⌵
  writeable="yes">
  <uei xmlns="">uei.opennms.org/nodes/nodeDown</uei>
  <rule xmlns="">IPADDR IPLIKE 10.10.12.*</rule>
  <destinationPath xmlns="">Email-Filial1 ⌵
    </destinationPath>
  <text-message xmlns="">
    All services are down on node %odelabel%.
  </text-message>
  <subject xmlns="">Node %odelabel% down.</subject>
```

```
<numeric-message xmlns="">111-%noticeid% ⌵
  </numeric-message>
</notification>
```

Описанное уведомление будет отправлено по адресу Email-Filial1 (см. файл destinationPaths.xml) при возникновении события nodeDown (<uei xmlns=""> uei.opennms.org/nodes/nodeDown </uei>), когда недоступны узлы, принадлежащие сети 10.10.12.0/24 (см. правило <rule xmlns="">IPADDR IPLIKE 10.10.12.*</rule>).

Аналогичным образом можно добавить уведомления, отправляемые другим адресатам при возникновении различных событий.

Тюнинг сбора и отображения SNMP-статистики

Настройки сбора информации, получаемой по протоколу SNMP, описываются в файле data-collection-config.xml. Данные собираются путем отправки GET-запросов, содержащих определенный OID (Object Identifier), устройству, поддерживающему протокол SNMP. В ответ возвращаются некоторые значения, которые сохраняются затем в RRD-файлах (RoundRobin Database (RRD) – это проект, выросший из известного проекта MRTG. Проект оказался довольно удачным и в настоящее время используется не только в openNMS, но и во многих других системах мониторинга).

В файле data-collection-config.xml уже содержится информация по OID (Object ID) из MIB (Management Information Base) наиболее распространенных устройств (как аппаратных, например Cisco Pix, так и программных – Asterisk), поддерживающих протокол SNMP.

OID описываются внутри тега <mibObj>:

```
<mibObj oid=".1.3.6.1.2.1.2.2.1.8" instance="ifIndex"
  alias="ifOperStatus" type="integer" />
```

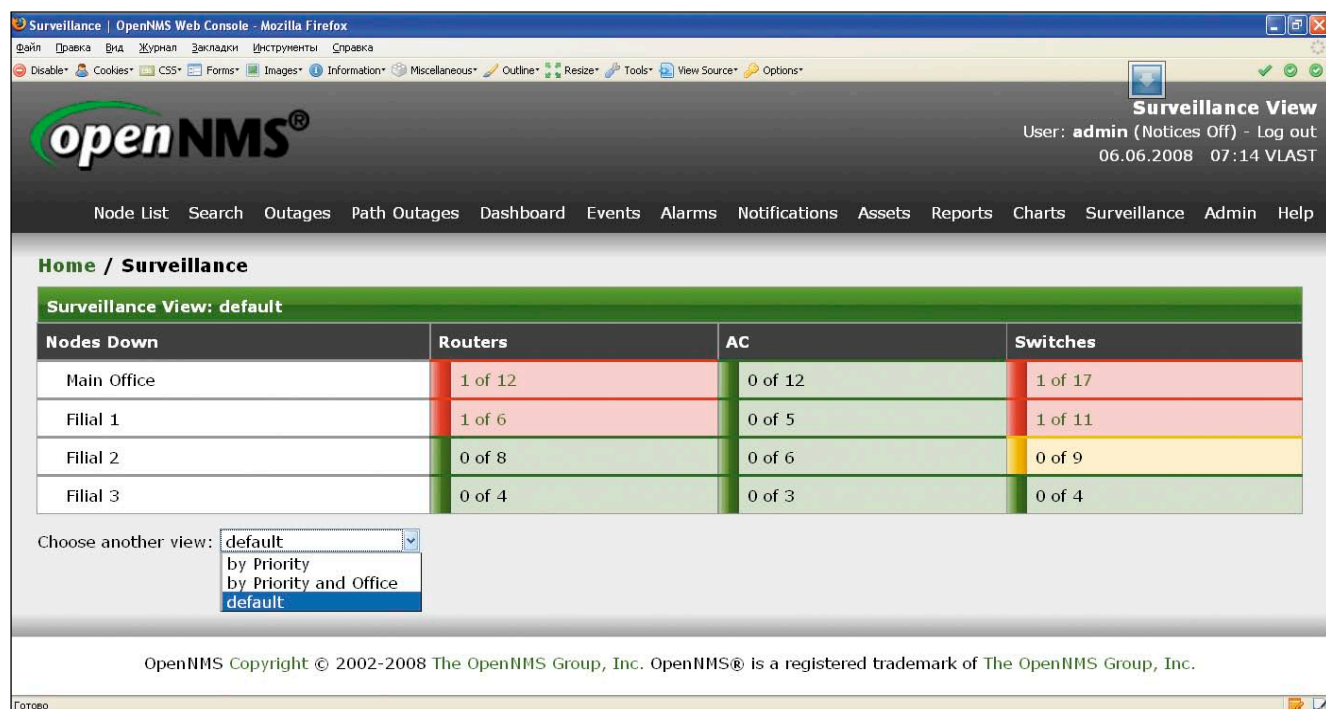


Рисунок 4. Вид меню «Surveillance» после настройки таблицы категорий

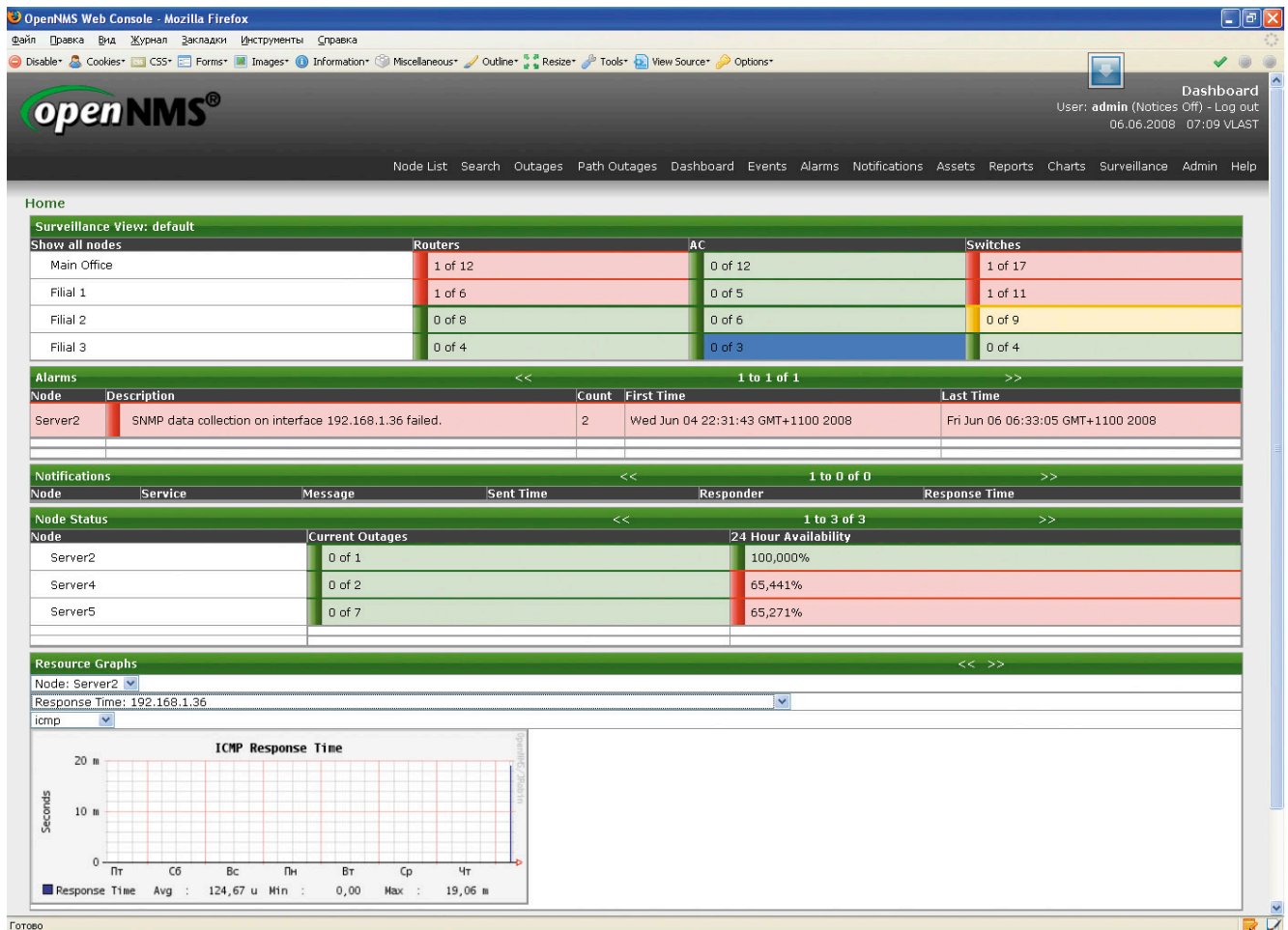


Рисунок 5. Меню «DashBoard» после настройки таблицы категорий

В свою очередь, для обеспечения более удобного дальнейшего использования OID группируются с помощью тега `<group>`, например:

```
<group name="my-mib2-interfaces" ifType="all">
  <mibObj oid=".1.3.6.1.2.1.2.2.1.8" instance="ifIndex"
    alias="ifOperStatus" type="integer" />
</group>
```

Если параметру `ifType` тега `<group>` присвоено значение «all», как в примере выше, это значит, что данные в группе имеют табличную природу, и тогда параметр `instance` тега `<mibObj>` принимает последовательные значения (в нашем случае `ifIndex`, содержащий индексы имеющихся в системе интерфейсов). Случай с нетабличными данными проще. Параметр `ifType` тега `<group>` принимает значение «ignore», а параметр `instance` тега `<mibObj>` принимает значение «0».

Различные типы SNMP-устройств описываются в тегах `<systemDef>`. Для каждого типа устройств определена одна или несколько OID-групп, на основе которых для устройства и его интерфейсов в `rrd`-файлы будет собираться статистика:

```
<systemDef name="Cisco Routers">
  <sysoidMask>.1.3.6.1.4.1.9.1.</sysoidMask>
  <collect>
    <includeGroup>cisco-memory</includeGroup>
    <includeGroup>cisco-router</includeGroup>
    <includeGroup>cisco-temperature</includeGroup>
```

```
<includeGroup>cisco-voltage</includeGroup>
<includeGroup>cisco-router-interface</includeGroup>
<includeGroup>mib2-interfaces</includeGroup>
<includeGroup>ads1-line</includeGroup>
</collect>
</systemDef>
```

При обнаружении на устройстве службы SNMP система OpenNMS пытается определить тип устройства, посылая GET-запрос с `OID=1.3.6.1.2.1.2.0`. Полученный ответ сравнивается со значением тега `<sysoidMask>`, вложенного в тег `<systemDef>`.

Например, для многочисленного семейства маршрутизаторов фирмы Cisco этот ответ будет содержать подстроку `.1.3.6.1.4.1.9.1`. На основании такого ответа SNMP-сборщик будет собирать информацию, посылая GET-запросы с `OID`, описанными в конфигурации для данного типа оборудования.

В случае наличия специфичного оборудования можно создать дополнительные группы `OID`, а также определить с помощью тега `<systemDef>` новый тип оборудования, в который и вложить необходимые `OID`-группы. Обычно с таким специфичным оборудованием распространяются и дополнительные `MIB`-базы. Для ускорения импортирования данных `OID` из таких `MIB`-файлов в состав дистрибутива `openNMS` включена утилита `mib2opennms`, с помощью которой можно преобразовать данные из `MIB`-файла в набор тегов `<mibObj>`, которые используются в файле `data-collection-config.xml`.

Проблема с кириллицей

Данная информация, возможно, уже потеряла актуальность в связи с выходом исправлений в новой версии 1.5.93. Дело в том, что в процессе работы с веб-интерфейсом в версии 1.5.90 обнаружилась одна неприятная проблема с отображением кириллицы. Для ее решения пришлось немного модифицировать некоторые *.jsp-файлы веб-интерфейса (в сервлетах такой проблемы не наблюдалось). Принцип модификации заключался в добавлении в JSP-страницу строки java-кода с указанием используемой кодировки UTF8:

```
<%response.setCharacterEncoding („UTF-8“);%>
```

То есть все данные, передаваемые сервером браузеру, должны отправляться в кодировке UTF8.

Возможно, существовало более элегантное решение проблемы отображения кириллицы, но на тот момент я его не нашел, поэтому до исправления данной ошибки воспользовался описанным выше способом. В последней версии (на момент написания статьи – 1.5.93) проблема с кириллицей полностью исправлена, что не может не радовать.

ле snmp-graph.properties. В качестве значений секции reports через запятую перечислены все доступные графики. При первоначальной установке OpenNMS их уже достаточно внушительное количество. После перечисления всех графиков находится полная информация по каждому из графиков. Чтобы лучше понять принцип построения графиков, можно добавить график состояния интерфейса во времени, принимающий значение «0» в случае «падения» интерфейса и «1» – в случае его нормального функционирования.

Для этого вернемся снова к файлу data-collection-config.xml, в который необходимо добавить соответствующий OID =1.3.6.1.2.1.2.2.1.8, например, в группу mib2-interfaces:

Давайте теперь рассмотрим, как записываются и хранятся в RRD-файлах собранные по протоколу SNMP данные. Периодичность сбора данных, а также продолжительность их хранения описывается в теге <rrd>:

```
<rrd step="60">
  <rra>RRA:AVERAGE:0.5:1:263520</rra>
  <rra>RRA:AVERAGE:0.5:60:8784</rra>
  <rra>RRA:AVERAGE:0.5:1440:366</rra>
  <rra>RRA:MAX:0.5:1440:366</rra>
  <rra>RRA:MIN:0.5:1440:366</rra>
</rrd>
```

Параметр step тега <rrd> определяет шаг хранения информации в rrd-файлах. Вложенный тег <rra> состоит из следующих атрибутов:

RRA:Cf:xff:steps:rows

где:

- **RRA** – однозначно определяет строку как конфигурационную команду.
- **cf** – тип объединения данных, принимает значения AVERAGE, MAX, MIN или LAST.
- **xff** – при объединении собранных значений в одно может случиться, что некоторые значения не определены (например, какое-то время узел был недоступен). Данный параметр определяет минимальный процент неопределенных данных, при котором все собранные за период данные становятся неопределенными. По умолчанию это 50% (0,5).
- **steps** – означает коэффициент группировки собранных данных, например:
 - ☑ 1 – данные сохраняются на каждом шаге (то есть ежеминутно, в случае, step=60);
 - ☑ 60 – данные группируются за 60 периодов и затем сохраняются (то есть интервал сохранения – 1 час).
- **rows** – определяет количество сохраняемых значений. Цифра 263520 означает, что данные, в случае ежеминутной группировки будут храниться около полугода (188 дней).

Собранная в RRD-файлах информация может быть представлена в виде графиков. Настройка графиков на основе собранных по SNMP данных производится в фай-

```
<group name="mib2-interfaces" ifType="all">
  <mibObj oid="1.3.6.1.2.1.2.2.1.8" instance="ifIndex"
    alias="ifOperStatus" type="integer" />
  <mibObj oid="1.3.6.1.2.1.2.2.1.10" instance="ifIndex"
    alias="ifInOctets" type="counter" />
  <mibObj oid="1.3.6.1.2.1.2.2.1.16" instance="ifIndex"
    alias="ifOutOctets" type="counter" />
  ... другие oid
</group>
```

Теперь для тех SNMP-устройств (точнее, для их интерфейсов, так как это данные уровня интерфейса, а не узла), в описании которых указана данная OID-группа, будет собираться статистика по состоянию интерфейса. Однако в графическом виде информацию о состоянии интерфейсов посмотреть пока не получится. Для просмотра графиков на основе rrd-данных необходимо будет внести соответствующие дополнения в конфигурационный файл snmp-graph.properties. Добавим строку mib2.opstat в секцию reports файла snmp-graph.properties. Теперь добавим следующие строки ниже секции report:

```
report.mib2.opstat.name=Interface Operational Status
report.mib2.opstat.columns=ifOperStatus
report.mib2.opstat.type=interfaceSnmp
report.mib2.opstat.command="--title="Interface \
  Operational Status" \
--lower-limit 0 --upper-limit 2 --rigid \
--vertical-label="Up(1)|Down(0)" \
--width 400 \
--height 100 \
DEF:opst={rrd1}:ifOperStatus:AVERAGE \
CDEF:copst=2,opst,- \
LINE2:copst#00ff00:"OperStatus" \
GPRINT:copst:AVERAGE:"Status (0 - down 1 - up) \
  \\\: %2.01f %s" \
```

В данных строках мы описали внешний вид графика, на котором будут отображаться данные о состоянии интерфейса (ноль на графике – интерфейс «опущен», а единица – интерфейс функционирует нормально).

В строке «report.mib2.opstat.columns=ifOperStatus» описываются данные, на основе которых строится график.

Значение interfaceSnmp параметра report.mib2.opstat.type означает, что график создается для интерфейсов узла (возможен тип nodeSNMP – то есть для всего узла, а не его интерфейсов).

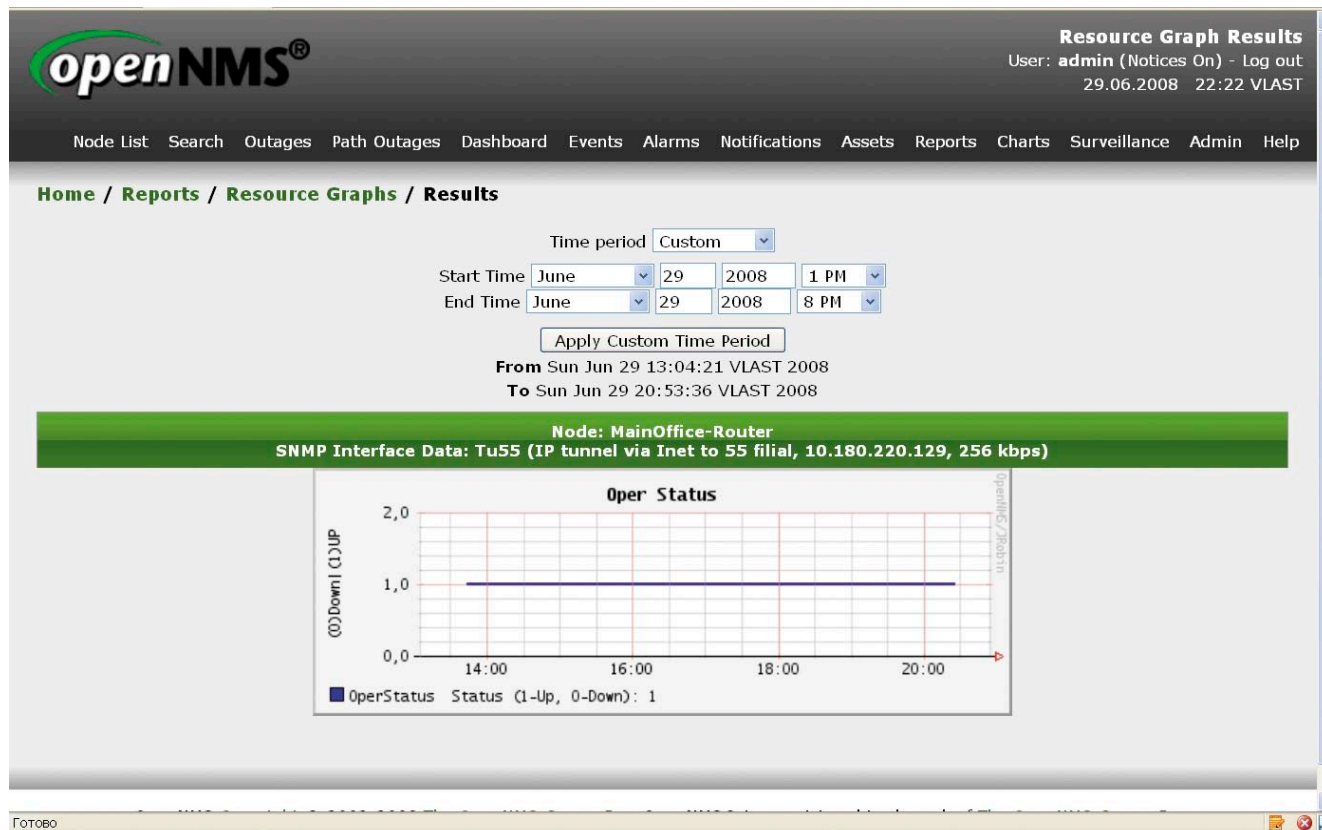


Рисунок 6. График изменения состояния (Up/Down) интерфейса во времени

В строке «DEF:opst={rrd1}:ifOperStatus:AVERAGE \» дается определение opst – переменной, на основе средних значений (AVERAGE) которой будет строиться график.

В строке «CDEF:copst=2,opst,- \» дано определение производной переменной copst, значение которой есть функция 2-opst. Функцию ввели для того, чтобы отображаемый график выглядел более привычно (в диапазоне значений от 0 до 1), так как возвращаемые значения для «поднятого» интерфейса – 1, в противном случае – 2. Функция 2 – opst решает данную проблему.

В строке «LINE2:copst#00ff00:"OperStatus" \» описывается вид графика, его цвет и название графика.

Строка «GPRINT:copst:AVERAGE:"Status (0 – down 1 – up) \\\: %2.0lf %s" \», как видно из названия, ответственна за рисование графика на основе переменной copst.

В строках height и width задаются размеры графика.

После внесения данной информации в конфигурацию openNMS мы можем просматривать на графике историю изменения оперативного состояния интерфейса (см. рис. 6).

Конечно, можно усомниться в практической полезности добавленного графика, но цель примера – показать, что аналогичным образом можно построить практически любой график на основе rrd-данных.

Заключение

Ну вот и закончено приведение в надлежащий вид установленной системы openNMS. Надеюсь, что полученных знаний о принципах работы системы, а также ее конфигурировании будет достаточно для дальнейшего самостоятельного освоения и использования.

Система оказалась очень гибкой и функциональной, хотя для настройки некоторых дополнительных функций и пришлось затратить некоторое количество усилий. В качестве несомненных плюсов также можно назвать и активность разработчиков проекта, с каждой новой версией добавляющих новые функции и исправляющих выявленные ошибки. За время, прошедшее после выхода первой статьи, было выпущено три обновления продукта, последнее из которых (версия 1.5.93 на момент написания статьи) было посвящено только работе над ошибками. Также на официальном сайте есть внятная документация, хотя, возможно, она не совсем оптимально структурирована. А после выхода статьи появилась возможность почитать о системе и на русском.

За кадром осталось описание еще многих возможностей, таких как интеграция с другими системами мониторинга, в том числе и коммерческими, распределенная установка системы и тонкости повышения производительности на высоконагруженных системах. Кроме того, данная система после некоторых настроек может дополняться графической картой устройств, что является удобной функцией. Но оставляю эти вопросы для изучения вам, коллеги, чтобы в повседневной работе оставались исследовательские и творческие моменты, а не только следование готовым мануалам и пошаговым инструкциям (что во многих случаях также полезно для повышения производительности труда). Удачи. 🍀

1. Сайт OpenNMS – <http://www.opennms.org>.
2. Документация OpenNMS – <http://www.opennms.org/index.php/Docu-overview>.