**Содержание:**

1. Мониторинг и анализ локальных сетей

2. Классификация средств мониторинга и анализа

2.1 Анализаторы протоколов

2.2 Сетевые анализаторы

3. Протокол SNMP

3.1 Отличия SNMPv3

3.2 Безопасность в SNMPv3

3.3 Недостатки протокола SNMP

**1. Мониторинг и анализ локальных сетей**

Постоянный контроль за работой локальной сети, составляющей основу любой корпоративной сети, необходим для поддержания ее в работоспособном состоянии. Контроль — это необходимый первый этап, который должен выполняться при управлении сетью. Ввиду важности этой функции ее часто отделяют от других функций систем управления и реализуют специальными средствами. Такое разделение функций контроля и собственно управления полезно для небольших и средних сетей, для которых установка интегрированной системы управления экономически нецелесообразна. Использование автономных средств контроля помогает администратору сети выявить проблемные участки и устройства сети, а их отключение или реконфигурацию он может выполнять в этом случае вручную. Процесс контроля работы сети обычно делят на два этапа — мониторинг и анализ.

На *этапе мониторинга* выполняется более простая процедура — процедура сбора первичных данных о работе сети: статистики о количестве циркулирующих в сети кадров и пакетов различных протоколов, состоянии портов концентраторов, коммутаторов и маршрутизаторов и т. п.

Далее выполняется этап *анализа,* под которым понимается более сложный и интеллектуальный процесс осмысления собранной на этапе мониторинга информации, сопоставления ее с данными, полученными ранее, и выработки предположений о возможных причинах замедленной или ненадежной работы сети.

Задачи мониторинга решаются программными и аппаратными измерителями, тестерами, сетевыми анализаторами, встроенными средствами мониторинга коммуникационных устройств, а также агентами систем управления. Задача анализа требует более активного участия человека и использования таких сложных средств, как экспертные системы, аккумулирующие практический опыт многих сетевых специалистов.

**2. Классификация средств мониторинга и анализа**

Все многообразие средств, применяемых для анализа и диагностики вычислительных сетей, можно разделить на несколько крупных классов.

* Агенты систем управления, поддерживающие функции одной из стандартных MIB (**MIB** (**Management Information Base**) — база данных информации управления, используемая в процессе управления сетью в качестве модели управляемого объекта в архитектуре [агент-менедже](http://ru.wikipedia.org/w/index.php?title=%D0%90%D0%B3%D0%B5%D0%BD%D1%82-%D0%BC%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80&action=edit&redlink=1)) и поставляющие информацию по протоколу SNMP или CMIP. Для получения данных от агентов обычно требуется наличие системы управления, собирающей данные от агентов в автоматическом режиме.
* Встроенные системы диагностики и управления (Embedded systems). Эти системы выполняются в виде программно-аппаратных модулей, устанавливаемых в коммуникационное оборудование, а также в виде программных модулей, встроенных в операционные системы. Они выполняют функции диагностики и управления только одним устройством, и в этом их основное отличие от централизованных систем управления. Примером средств этого класса может служить модуль управления многосегментным повторителем Ethernet, реализующий функции автосегментации портов при обнаружении неисправностей, приписывания портов внутренним сегментам повторителя и некоторые другие. Как правило, встроенные модули управления «по совместительству» выполняют роль SNMP-агентов, поставляющих данные о состоянии устройства для систем управления.
* Анализаторы протоколов (Protocol analyzers). Представляют собой программные или аппаратно-программные системы, которые ограничиваются в отличие от систем управления лишь функциями мониторинга и анализа трафика в сетях. Хороший анализатор протоколов может захватывать и декодировать пакеты большого количества протоколов, применяемых в сетях, — обычно несколько десятков. Анализаторы протоколов позволяют установить некоторые логические условия для захвата отдельных пакетов и выполняют полное декодирование захваченных пакетов, то есть показывают в удобной для специалиста форме вложенность пакетов протоколов разных уровней друг в друга с расшифровкой содержания отдельных полей каждого пакета.
* Экспертные системы. Этот вид систем аккумулирует знания технических специалистов о выявлении причин аномальной работы сетей и возможных способах приведения сети в работоспособное состояние. Экспертные системы часто реализуются в виде отдельных подсистем различных средств мониторинга и анализа сетей: систем управления сетями, анализаторов протоколов, сетевых анализаторов. Простейшим вариантом экспертной системы является контекстно-зависимая система помощи. Более сложные экспертные системы представляют собой, так называемые базы знаний, обладающие элементами искусственного интеллекта. Примерами таких систем являются экспертные системы, встроенные в систему управления Spectrum компании Cabletron и анализатора протоколов Sniffer компании Network General. Работа экспертных систем состоит в анализе большого числа событий для выдачи пользователю краткого диагноза о причине неисправности сети.
* Оборудование для диагностики и сертификации кабельных систем. Условно это оборудование можно поделить на четыре основные группы: сетевые мониторы, приборы для сертификации кабельных систем, кабельные сканеры и тестеры.
  + Сетевые мониторы (называемые также сетевыми анализаторами) предназначены для тестирование кабелей различных категорий. Сетевые мониторы собирают также данные о статистических показателях трафика — средней интенсивности общего трафика сети, средней интенсивности потока пакетов с определенным типом ошибки и т. п. Эти устройства являются наиболее интеллектуальными устройствами из всех четырех групп устройств данного класса, так как работают не только на физическом, но и на канальном, а иногда и на сетевом уровнях.
  + Устройства для сертификации кабельных систем выполняют сертификацию в соответствии с требованиями одного из международных стандартов на кабельные системы.
  + Кабельные сканеры используются для диагностики медных кабельных систем.
  + Тестеры предназначены для проверки кабелей на отсутствие физического разрыва. Многофункциональные портативные устройства анализа и диагностики. В связи с развитием технологии больших интегральных схем появилась возможность производства портативных приборов, которые совмещали бы функции нескольких устройств: кабельных сканеров, сетевых мониторов и анализаторов протоколов.

**2.1 Анализаторы протоколов**

Анализатор протоколов представляет собой либо специализированное устройство, либо персональный компьютер, обычно переносной, класса Notebook, оснащенный специальной сетевой картой и соответствующим программным обеспечением.

Применяемые сетевая карта и программное обеспечение должны соответствовать технологии сети (Ethernet, Token Ring, FDDI, Fast Ethernet). Анализатор подключается к сети точно так же, как и обычный узел. Отличие состоит в том, что анализатор может принимать все пакеты данных, передаваемые по сети, в то время как обычная станция — только адресованные ей. Для этого сетевой адаптер анализатора протоколов переводится в режим *«беспорядочного» захвата —promiscuousmode.*

Программное обеспечение анализатора состоит из ядра, поддерживающего работу сетевого адаптера и программного обеспечения, декодирующего протокол канального уровня, с которым работает сетевой адаптер, а также наиболее распространенные протоколы верхних уровней, например IP, TCP, ftp, telnet, HTTP, IPX, NCP, NetBEUI, DECnet и т. п. В состав некоторых анализаторов может входить также экспертная система, которая позволяет выдавать пользователю рекомендации о том, какие эксперименты следует проводить в данной ситуации, что могут означать те или иные результаты измерений, как устранить некоторые виды неисправности сети.

Анализаторы протоколов имеют некоторые общие свойства.

* Возможность (кроме захвата пакетов) измерения среднестатистических показателей трафика в сегменте локальной сети, в котором установлен сетевой адаптер анализатора.
* Обычно измеряется коэффициент использования сегмента, матрицы перекрестного трафика узлов, количество хороших и плохих кадров, прошедших через сегмент.
* Возможность работы с несколькими агентами, поставляющими захваченные пакеты из разных сегментов локальной сети. Эти агенты чаще всего взаимодействуют с анализатором протоколов по собственному протоколу прикладного уровня, отличному от SNMP или CMIP.
* Наличие развитого графического интерфейса, позволяющего представить результаты декодирования пакетов с разной степенью детализации.
* Фильтрация захватываемых и отображаемых пакетов. Условия фильтрации задаются в зависимости от значения адресов назначения и источника, типа протокола или значения определенных полей пакета. Пакет либо игнорируется, либо записывается в буфер захвата. Использование фильтров значительно ускоряет и упрощает анализ, так как исключает захват или просмотр ненужных в данный момент пакетов.
* Использование триггеров. Триггеры — это задаваемые администратором некоторые условия начала и прекращения процесса захвата данных из сети. Такими условиями могут быть: время суток, продолжительность процесса захвата, появление определенных значений в кадрах данных. Триггеры могут использоваться совместно с фильтрами, позволяя более детально и тонко проводить анализ, а также продуктивнее расходовать ограниченный объем буфера захвата.
* Многоканальность. Некоторые анализаторы протоколов позволяют проводить одновременную запись пакетов от нескольких сетевых адаптеров, что удобно для сопоставления процессов, происходящих в разных сегментах сети.

Возможности анализа проблем сети на физическом уровне у анализаторов протоколов минимальные, поскольку всю информацию они получают от стандартных сетевых адаптеров.

Поэтому они передают и обобщают информацию физического уровня, которую сообщает им сетевой адаптер, а она во многом зависит от типа сетевого адаптера.

Некоторые сетевые адаптеры сообщают более детальные данные об ошибках кадров и интенсивности коллизий в сегменте, а некоторые вообще не передают такую информацию верхним уровням протоколов, на которых работает анализатор протоколов.

С распространением серверов Windows NT все более популярным становится анализатор Network Monitor фирмы Microsoft. Он является частью сервера управления системой SMS, а также входит в стандартную поставку Windows NT Server, начиная с версии 4.0 (версия с усеченными функциями). Network Monitor в версии SMS является многоканальным анализатором протоколов, поскольку может получать данные от нескольких агентов Network Monitor Agent, работающих в среде Windows NT Server, однако в каждый момент времени анализатор может работать только с одним агентом, так что сопоставить данные разных каналов с его помощью не удастся. Network Monitor поддерживает фильтры захвата (достаточно простые) и дисплейные фильтры, отображающие нужные кадры после захвата (более сложные). Экспертной системой Network Monitor не располагает.

**2.2 Сетевые анализаторы**

Сетевые анализаторы представляют собой эталонные измерительные приборы для диагностики и сертификации кабелей и кабельных систем. Они могут с высокой точностью измерить все электрические параметры кабельных систем, а также работают на более высоких уровнях стека протоколов. Сетевые анализаторы генерируют синусоидальные сигналы в широком диапазоне частот, что позволяет измерять на приемной паре амплитудно-частотную характеристику и перекрестные наводки, затухание и суммарное затухание. Сетевой анализатор представляет собой лабораторный прибор больших размеров, достаточно сложный в обращении.

Многие производители дополняют сетевые анализаторы функциями статистического анализа трафика — коэффициента использования сегмента, уровня широковещательного трафика, процента ошибочных кадров, а также функциями анализатора протоколов, которые обеспечивают захват пакетов разных протоколов в соответствии с условиями фильтров и декодирование пакетов.

**Сетевая статистика**

В этой группе собраны наиболее важные статистические показатели — коэффициент использования сегмента (utilization), уровень коллизий, уровень ошибок и уровень широковещательного трафика. Превышение этими показателями определенных порогов в первую очередь говорят о проблемах в том сегменте сети, к которому подключен многофункциональный прибор.

**Статистика ошибочных кадров**

Эта функция позволяет отслеживать все типы ошибочных кадров для определенной технологии. Например, для технологии Ethernet характерны следующие типы ошибочных кадров.

* Укороченные кадры (Short frames). Это кадры, имеющие длину, меньше допустимой, то есть меньше 64 байт. Иногда этот тип кадров дифференцируют на два класса — просто короткие кадры (short), у которых имеется корректная контрольная сумма, и «коротышки» (runts), не имеющие корректной контрольной суммы. Наиболее вероятными причинами появления укороченных кадров являются неисправные сетевые адаптеры и их драйверы.
* Удлиненные кадры (Jabbers). Это кадры, имеющие длину, превышающую допустимое значение в 1518 байт с хорошей или плохой контрольной суммой. Удлиненные кадры являются следствием затянувшейся передачи, которая появляется из-за неисправностей сетевых адаптеров.
* Кадры нормальных размеров, но с плохой контрольной суммой (Bad FCS) и кадры с ошибками выравнивания по границе байта. Кадры с неверной контрольной суммой являются следствием множества причин — плохих адаптеров, помех на кабелях, плохих контактов, некорректно работающих портов повторителей, мостов, коммутаторов и маршрутизаторов. Ошибка выравнивания всегда сопровождается ошибкой по контрольной сумме, поэтому некоторые средства анализа-трафика не делают между ними различий. Ошибка выравнивания может быть следствием прекращения передачи кадра при распознавании коллизии передающим адаптером.
* Кадры-призраки (ghosts) являются результатом электромагнитных наводок на кабеле. Они воспринимаются сетевыми адаптерами как кадры, не имеющие нормального признака начала кадра — 10101011. Кадры-призраки имеют длину более 72 байт, в противном случае они классифицируются как удаленные коллизии. Количество обнаруженных кадров-призраков в большой степени зависит от точки подключения сетевого анализатора. Причинами их возникновения являются петли заземления и другие проблемы с кабельной системой.

Знание процентного распределения общего количества ошибочных кадров по их типам может многое подсказать администратору о возможных причинах неполадок в сети. Даже небольшой процент ошибочных кадров может привести к значительному снижению полезной пропускной способности сети, если протоколы, восстанавливающие искаженные кадры, работают с большими тайм-аутами ожидания квитанций. Считается, что в нормально работающей сети процент ошибочных кадров не должен превышать 0,01 %, то есть не более 1 ошибочного кадра из 10 000.

**Статистика по коллизиям**

Эта группа характеристик дает информацию о количестве и видах коллизий, отмеченных на сегменте сети, позволяет определить наличие и местонахождение проблемы. Анализаторы протоколов обычно не могут дать дифференцированной картины распределения общего числа коллизий по их отдельным типам, в то же время знание преобладающего типа коллизий может помочь понять причину плохой работы сети.

Ниже приведены основные типы коллизий сети Ethernet.

* Локальная коллизия (Local Collision). Является результатом одновременной передачи двух или более узлов, принадлежащих к тому сегменту, в котором производятся измерения. Если многофункциональный прибор не генерирует кадры, то в сети на витой паре или волоконно-оптическом кабеле локальные коллизии не фиксируются. Слишком высокий уровень локальных коллизий является следствием проблем с кабельной системой.
* Удаленная коллизия (Remote Collision). Эти коллизии происходят на другой стороне повторителя (по отношению к тому сегменту, в котором установлен измерительный прибор). В сетях, построенных на многопортовых повторителях (10Base-T, 10Base-FL/FB, 100Base-TX/FX/T4, Gigabit Ethernet), все измеряемые коллизии являются удаленными (кроме тех случаев, когда анализатор сам генерирует кадры и может быть виновником коллизии). Не все анализаторы протоколов и средства мониторинга одинаковым образом фиксируют удаленные коллизии. Это происходит из-за того, что некоторые измерительные средства и системы не фиксируют коллизии, происходящие при передаче преамбулы.
* Поздняя коллизия (Late Collision). Это коллизия, которая происходит после передачи первых 64 байт кадра (по протоколу Ethernet коллизия должна обнаруживаться при передаче первых 64 байт кадра). Результатом поздней коллизии будет кадр, который имеет длину более 64 байт и содержит неверное значение контрольной суммы. Чаще всего это указывает на то, что сетевой адаптер, являющийся источником конфликта, оказывается не в состоянии правильно прослушивать линию и поэтому не может вовремя остановить передачу. Другой причиной поздней коллизии является слишком большая длина кабельной системы или слишком большое количество промежуточных повторителей, приводящее к превышению максимального значения времени двойного оборота сигнала. Средняя интенсивность коллизий в нормально работающей сети должна быть меньше 5 %. Большие всплески (более 20 %) могут быть индикатором кабельных проблем.

**Распределение используемых сетевых протоколов**

Эта статистическая группа относится к протоколам сетевого уровня. На дисплее отображается список основных протоколов в убывающем порядке относительно процентного соотношения кадров, содержащих пакеты данного протокола к общему числу кадров в сети.

**Основные отправители (Top Sendes)**

Функция позволяет отслеживать наиболее активные передающие узлы локальной сети. Прибор можно настроить на фильтрацию по единственному адресу и выявить список основных отправителей кадров для данной станции. Данные отражаются на дисплее в виде диаграммы вместе с перечнем основных отправителей кадров.

**Основные получотели (Top Receivers)**

Функция позволяет следить за наиболее активными узлами-получателями сети. Информация отображается в виде, аналогичном приведенному выше.

**Основные генераторы широковещательного трафика (Top Broadcasters)**

Функция выявляет станции сети, которые больше остальных генерируют кадры с широковещательными и групповыми адресами.

**Генерирование трафика (Traffic Generation)**

Прибор может генерировать трафик для проверки работы сети при повышенной нагрузке. Трафик может генерироваться параллельно с активизированными функциями *Сетевая статистика, Статистика ошибочных кадров* и *Статистика по коллизиям.*

Пользователь может задать параметры генерируемого трафика, такие как интенсивность и размер кадров. Для тестирования мостов и маршрутизаторов прибор может автоматически создавать заголовки IP- и IPX-пакетов, и все что требуется от оператора — это внести адреса источника и назначения.

В ходе испытаний пользователь может увеличить на ходу размер и частоту следования кадров с помощью клавиш управления курсором. Это особенно ценно при поиске источника проблем производительности сети и условий возникновения отказов.

**3. Протокол SNMP**

Всю необходимую информацию протокол SNMP получает из базы управляющей информации (ManagementInformationBase, MIB). MIB представляет собой базу данных стандартизированной структуры. База данных имеет древовидную структуру, а все переменные классифицированы по тематике. Каждое поддерево содержит определенную тематическую подгруппу переменных. Наиболее важные компоненты, отвечающие за работу сетевых узлов, объединены в подгруппе MIB-II.

Существуют два типа MIB: стандартные и фирменные. Стандартные MIB определены комиссией по деятельности Интернет (Internet Activity Board, IAB), а фирменные - производителем устройства.

В таблице 1 приведён список наиболее распространенных стандартов баз управляющей информации.

Таблица 1

|  |  |
| --- | --- |
| База | Назначение |
| MIB-II | Задает множество объектов, которые могут быть использованы для управления сетевыми интерфейсами. |
| MIB повторителя | Включена в подмножество MIB-II. Устанавливает объекты, которые можно использовать для управления повторителем. |
| MIB моста | Включена в подмножество MIB-II.  Задает объекты данных, которые можно использовать для управления мостом. |
| RMON MIB | Указывает объекты данных, которые можно использовать для управления сетью в целом, при помощи протокола RMON. |

В базах данных, указанных в таблице 1, присутствует множество переменных, которые могут быть полезны для диагностирования сети и сетевых устройств.

Например, используя MIB-II, можно получить сведения об общем количестве пакетов, переданных сетевым интерфейсом, а с помощью MIB повторителя можно узнать информацию о количестве коллизий в порту.

В MIB каждый объект имеет имя и тип. Имя объекта характеризует его положение в дереве MIB. При этом имя дочернего узла включает в себя имя родительского узла и задается целым числом.

**3.1 Отличия SNMPv3**

SNMP – протокол прикладного уровня. Он предназначен для обмена информацией между сетевыми устройствами. При помощи этого протокола, сетевой администратор может производить анализ сетевого оборудования, находить и решать множество сетевых проблем.

В Декабре 1997 года с выходом SNMPv3, пользователям стали доступны новые службы, такие как: ограничение доступа, защита данных и аутентификация пользователя*.*

Кроме этого, стоит отметить, что SNMPv3 перенял модульную архитектуру от своих предшественников. Это обеспечивает поддержку предыдущих версий SNMPи, не смотря на то, что SNMPv1 и SNMPv2 не поддерживают аутентификацию и шифрование, у Вас будет возможность управления устройствами, которые поддерживают эти версии.

 При создании новой версии разработчики руководствовались следующими принципами:

1. необходимо обеспечить большую безопасность протокола (особенно для операций типа SET);
2. SNMPv3 должен иметь возможность дальнейшего развития и расширения;
3. протокол должен остаться простым и понятным;
4. настройки параметров безопасности SNMPv3 должны быть максимально простыми;

 В SNMPv3 уже не применяются термины «агент» и «менеджер», теперь используются термины «сущности». Как и раньше одна сущность находится на управляемом устройстве, а вторая занимается опросом приложений.

 У сущностей-агентов и сущностей-менеджеров теперь есть ядро, которое выполняет четыре основные функции (см. Рисунок 1):

1.функции диспетчера;

2.обработка сообщений;

3.функции безопасности;

4.контроль доступа.

***Диспетчер*** – это простая система управления входящим и исходящим трафиком. Для каждого исходящего блока данных (PDU) он определяет тип необходимой обработки (SNMPv1, SNMPv2, SNMPv3) и передает блок данных соответствующему модулю в системе обработки сообщений.

После того как система обработки сообщений вернет сообщение, которое содержит этот блок данных, Диспетчер отправит его на транспортный уровень для последующей передачи. Для входящих сообщений, Диспетчер проводит обратную операцию.

***Система обработки сообщений*** получает от Диспетчера исходящие блоки данных (PDU), добавляет к ним подходящий заголовок и возвращает их обратно Диспетчеру.

***Система безопасности*** отвечает за шифрование и аутентификацию. Все исходящие сообщения перед отправкой сначала передаются из системы обработки сообщений в систему безопасности, где все шифруются поля в заголовке сообщения, блок данных (PDU), генерируется код аутентификации и добавляется к заголовку сообщения.

После этого сообщение передается обратно в систему обработки сообщений. Точно такая же операция, но в обратном порядке производится для всех входящих сообщений.

***Система контроля доступа*** управляет службами аутентификации для контроля доступа к MIB исходя из содержимого блоков данных. (PDU). Теоретически, система контроля доступа может работать с самыми разными моделями контроля доступа, но на данный момент в RFC 2275 описана только одна модель – VACM (View-BasedAccessControlModel)

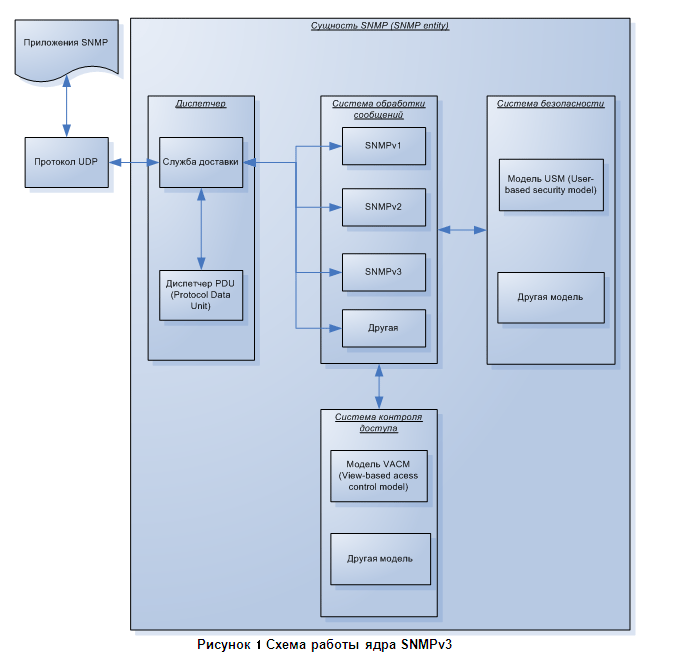
Таблица 2 - Основные методы SNMP

|  |  |  |
| --- | --- | --- |
| Метод | Для чего применяется | Поддерживается |
| GET | Используется менеджером для получения данных из MIB. Размер сообщения ограничен возможностями агента. | SNMPv1-3 |
| GET-NEXT | Метод позволяет последовательно выполнить набор команд иполучить набор значений из MIB | SNMPv1-3 |
| GET-BULK | Используется менеджером для получения сразу большого количества данных из MIB. Размер сообщения отсылаемого агентом не ограничен. | SNMPv2, SNMPv3 |
| SET | Используется менеджером для установки значений в MIB агента | SNMPv1-3 |
| GET-RESPONSE |  | SNMPv1-3 |
| TRAP | Используется агентом чтобы послать сигнал менеджеру | SNMPv1-3 |
| NOTIFICATION |  | SNMPv2, SNMPv3 |
| INFORM | Используется менеджером для отсылки сигнала другому менеджеру | SNMPv2, SNMPv3 |
| REPORT |  | SNMPv2, SNMPv3 |

 При помощи этих команд и стандартной базы MIB можно получить самую разнообразную информацию.

Например: количество принятых и отправленных пакетов по TCP, IP, UDP или ICMP. А еще можно узнать о количестве ошибок, которые были обнаружены во времяотправки или получения пакетов.

 При разработке SNMPv3 немало внимания было уделено безопасности протокола. Теперь стала поддерживаться модель, ориентированная на пользователя (User-BasedSecurityModel сокр. USM*<\* см. RFC 3414>*) благодаря которой стало возможным добавление модулей аутентификации и шифрованиябез смены базовой архитектуры.



**3.2 Безопасность в SNMPv3**

Модель USM включает в себя модуль аутентификации, модуль шифрования и модуль контроля времени. При этом, модуль аутентификации и шифрования занимаются защитой данных, а модуль контроля времени синхронизирует время между сущностями SNMP.

 Основные проблемы, которые необходимо было решить при помощи модели USM:

 Изменение данных сущностями не прошедшими аутентификацию;

1. Возможность откладывания каких-либо действий на неопределенное время или повторение одних и тех же действий с произвольными интервалами;
2. Возможность заблокировать обмен данными между сущностями;
3. Возможность перехвата трафика при передаче между сущностями;
4. Возможность «маскарада», т.е. сущность не прошедшая аутентификацию, могла прикинуться сущностью прошедшей аутентификацию.

 Проблему решили следующим образом: для каждого сетевого устройства пароль преобразуется в некоторый уникальный ключ. Это обеспечивает дополнительную безопасность т.к. даже в том случае, если ключ будет перехвачен, злоумышленник получит доступ только к одному сетевому устройству. Для шифрования пароля используется алгоритм MD5, но разработчики видимо решили, что это не обеспечит достаточной сохранности пароля и поэтому блок PDU дважды хэшируется при помощи двух разных ключей, которые в свою очередь генерируются из закрытого ключа. Позже, первые 12 октетов используются как код аутентификации сообщения, который добавляется к сообщению. Такой же процесс приходится производить на другой стороне, но только в обратном порядке. Несмотря на всю сложность и энергоемкость процесса передачи данных между сущностями SNMP, по мнению разработчиков, алгоритм шифрования (DES) на самом деле не обеспечивает достаточной защиты информации, поэтому в дальнейшем предполагается использовать другие алгоритмы. Например, алгоритм Диффи-Хиллмана (Diffie-Hillman)

 Разработчиками предусмотрено 3 уровня безопасности:

1. noAuthNoPriv – пароли передаются в открытом виде, конфиденциальность данных отсутствует.
2. authNoPriv – аутентификация без конфиденциальности. Большинство пользователейиспользует именно этот уровень т.к уровень защищенности в нем уже достаточно высок, а сетевые устройства не перегружаются шифрованием данных.
3. authPriv – аутентификация и шифрование. Максимальный уровень защищенности.

 Как правило, покупатели сначала выбирают второй уровень безопасности и лишь немногие из них, потом начинают использовать третий. Одной из причин, по которой не используется третий уровень, является то, что он перегружает сетевые устройства.

На данный момент закончена разработка новой спецификации DataOverCableServiceInterfaceSpecification*<см. стандарт RFC 3256>*, а для управления ключами многие пользователи уже используют алгоритмы Диффи-Хиллмана (Diffie-Hillman) и Kerberosвместо DES.Скорее всего, это означает, что скоро можно будет ожидать выход новой версии протокола SNMP.

Интернет - гигантская сеть. Напрашивается вопрос, как она сохраняет свою целостность и функциональность без единого управления? Если же учесть разнородность ЭВМ, маршрутизаторов и программного обеспечения, используемых в сети, само существование Интернет представится просто чудом. Так как же решаются проблемы управления в Интернет? Отчасти на этот вопрос уже дан ответ - сеть сохраняет работоспособность за счет жесткой протокольной регламентации. "Запас прочности" заложен в самих протоколах. Функции диагностики возложены, как было сказано выше, на протокол ICMP. Учитывая важность функции управления, для этих целей создано два протокола SNMP (Simple Network Management Protocol, RFC-1157, -1215, -1187, -1089, **std-15** разработан в 1988 году) и CMOT (Common Management Information services and protocol over TCP/IP, RFC-1095, в последнее время применение этого протокола ограничено). Обычно управляющая прикладная программа воздействует на сеть по цепочке SNMP-UDP-IP-Ethernet. Наиболее важным объектом управления обычно является внешний порт сети (gateway) или маршрутизатор сети. Каждому управляемому объекту присваивается уникальный идентификатор.

Протокол SNMP работает на базе транспортных возможностей UDP (возможны реализации и на основе ТСР) и предназначен для использования сетевыми управляющими станциями. Он позволяет управляющим станциям собирать информацию о положении в сети Интернет. Протокол определяет формат данных, а их обработка и интерпретация остаются на усмотрение управляющих станций или менеджера сети. SNMP-сообщения не имеют фиксированного формата и фиксированных полей. При своей работе SNMP использует управляющую базу данных (MIB - management information base, RFC-1213, -1212, **std-17**).

Алгоритмы управления в Интернет обычно описывают в нотации ASN.1 (Abstract Syntax Notation). Все объекты в Интернет разделены на 10 групп и описаны в MIB: система, интерфейсы, обмены, трансляция адресов, IP, ICMP, TCP, UDP, EGP, SNMP. В группу "система" входит название и версия оборудования, операционной системы, сетевого программного обеспечения и пр.. В группу "интерфейсы" входит число поддерживаемых интерфейсов, тип интерфейса, работающего под IP (Ethernet, LAPB etc.), размер дейтограмм, скорость обмена, адрес интерфейса. IP-группа включает в себя время жизни дейтограмм, информация о фрагментации, маски субсетей и т.д. В TCP-группу входит алгоритм повторной пересылки, максимальное число повторных пересылок и пр.. Ниже приведена таблица (3) команд (pdu - protocol data unit) SNMP:

Таблица 3 - Команды SNMP

|  |  |  |
| --- | --- | --- |
| Команда SNMP | Тип PDU | Назначение |
| GET-request | 0 | Получить значение указанной переменной или информацию о состоянии сетевого элемента; |
| GET\_next\_request | 1 | Получить значение переменной, не зная точного ее имени (следующий логический идентификатор на дереве MIB); |
| SET-request | 2 | Присвоить переменной соответствующее значение. Используется для описания действия, которое должно быть выполнено; |
| GET response | 3 | Отклик на GET-request, GET\_next\_request и SET-request. Содержит также информацию о состоянии (коды ошибок и другие данные); |
| TRAP | 4 | Отклик сетевого объекта на событие или на изменение состояния. |
| GetBulkRequest | 5 | Запрос пересылки больших объемов данных, например, таблиц. |
| InformRequest | 6 | Менеджер обращает внимание партнера на определенную информацию в MIB. |
| SNMPv3-Trap | 7 | Отклик на событие (расширение по отношению v1 и v2). |
| Report | 8 | Отчет (функция пока не задана). |

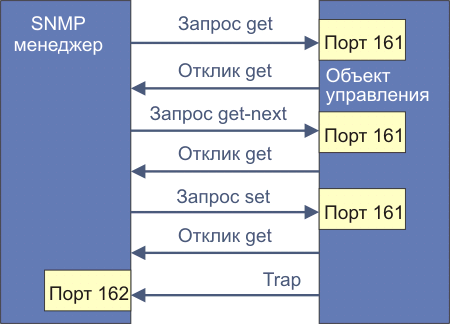


Рис. 2 - Схема запросов/откликов SNMP

Формат SNMP-сообщений, вкладываемых в UDP-дейтограммы, имеет вид (рис. 4.4.13.2):

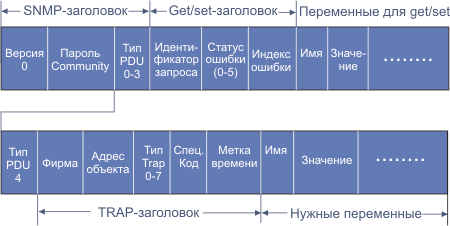


Рис. 3 - Формат SNMP-сообщений, вкладываемых в UDP-дейтограммы

Поле *версия* содержит значение, равное номеру версии SNMP минус один. Поле *пароль* (community - определяет группу доступа) содержит последовательность символов, которая является пропуском при взаимодействии менеджера и объекта управления. Обычно это поле содержит 6-байтовую строку public, что означает общедоступность. Для запросов GET, GET-next и SET значение *идентификатора запроса* устанавливается менеджером и возвращается объектом управления в отклике GET, что позволяет связывать в пары запросы и отклики. Поле *фирма* (enterprise) = sysobjectid объекта. Поле *статус ошибки* характеризуется целым числом, присланным объектом управления:

Таблица 4. Номера и назначения используемых портов

|  |  |  |
| --- | --- | --- |
| Назначение | Порт | Пояснение |
| SNMP | 161/TCP | Simple Network Management Protocol |
| SNMP | 162/TCP | Trap |
| SMUX | 199/TCP | SNMP Unix Multiplexer |
| SMUX | 199/UDP | SNMP Unix Multiplexer |
| synoptics-relay | 391/TCP | SynOptics SNMP Relay Port |
| synoptics-relay | 391/UDP | SynOptics SNMP Relay Port |
| Agentx | 705/TCP | AgentX |
| snmp-tcp-port | 1993/TCP | cisco SNMP TCP port |
| snmp-tcp-port | 1993/UDP | cisco SNMP TCP port |

Таблица 5 - Коды ошибок

|  |  |  |
| --- | --- | --- |
| Статус ошибки | Имя ошибки | Описание |
| 0 | Noerror | Все в порядке; |
| 1 | Toobig | Объект не может уложить отклик в одно сообщение; |
| 2 | Nosuchname | В операции указана неизвестная переменная; |
| 3 | badvalue | В команде set использована недопустимая величина или неправильный синтаксис; |
| 4 | Readonly | Менеджер попытался изменить константу; |
| 5 | Generr | Прочие ошибки. |

Если произошла ошибка, поле *индекс ошибки* (*error index)* характеризует, к какой из переменных это относится. **error index** является указателем переменной и устанавливается объектом управления не равным нулю для ошибок badvalue, readonly и nosuchname. Для оператора TRAP (тип PDU=4) формат сообщения меняется. Таблица типов TRAPпредставлена ниже (4.4.13.4):

Таблица 6 - Коды TRAP

|  |  |  |
| --- | --- | --- |
| Тип TRAP | Имя TRAP | Описание |
| 0 | Coldstart | Установка начального состояния объекта. |
| 1 | Warmstart | Восстановление начального состояния объекта. |
| 2 | Linkdown | Интерфейс выключился. Первая переменная в сообщении идентифицирует интерфейс. |
| 3 | Linkup | Интерфейс включился. Первая переменная в сообщении идентифицирует интерфейс. |
| 4 | Authenticationfailure | От менеджера получено snmp-сообщение с неверным паролем (community). |
| 5 | EGPneighborloss | R$GP-партнер отключился. Первая переменная в сообщении определяет IP-адрес партнера. |
| 6 | Entrprisespecific | Информация о TRAP содержится в поле специальный код. |

Для *тип TRAP* 0-4 поле *специальный код* должно быть равно нулю. Поле *временная метка* содержит число сотых долей секунды (число тиков) с момента инициализации объекта управления. Так прерывание coldstart выдается объектом через 200 мс после инициализации.

В последнее время широкое распространение получила идеология распределенного протокольного интерфейса DPI (Distributed Protocol Interface). Для транспортировки snmp-запросов может использоваться не только UDP-, но и TCP-протокол. Это дает возможность применять SNMP-протокол не только в локальных сетях. Форматы SNMP-DPI-запросов (версия 2.0) описаны в документе RFC-1592. Пример заголовка snmp-запроса (изображенные поля образуют единый массив; см. рис. 4.4.13.3):

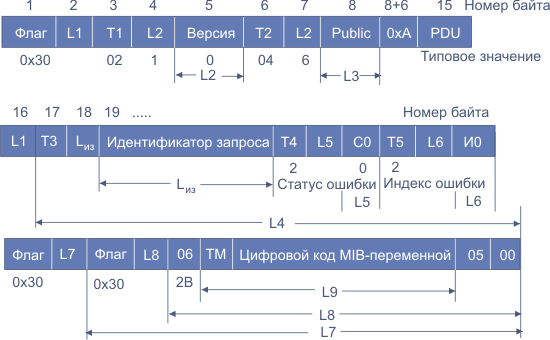


Рис. 4 - Формат заголовка SNMP-запроса

Поле *Флаг*=0x30 является признаком ASN.1-заголовка. Коды Ln - представляют собой длины полей, начинающиеся с байта, который следует за кодом длины, вплоть до конца сообщения-запроса (n - номер поля длины), если не оговорено другое. Так L1 - длина пакета-запроса, начиная с T1 и до конца пакета, а L3 - длина поля пароля. Субполя Tn - поля типа следующего за ними субполя запроса. Так T1=2 означает, что поле характеризуется целым числом, а T2=4 указывает на то, что далее следует пароль (поле community, в приведенном примере = public). Цифры под рисунками означают типовые значения субполей. Код 0xA - является признаком GET-запроса, за ним следует поле кода PDU (=0-4, см. табл. 4.4.13.1)

Блок субполей *идентификатора запроса* служит для тех же целей, что и другие идентификаторы - для определения пары запрос-отклик. Собственно идентификатор запроса может занимать один или два байта, что определяется значением Lиз. СО - статус ошибки (СО=0 - ошибки нет); ТМ - тип MIB-переменной (в приведенном примере = 0x2B); ИО - индекс ошибки. Цифровой код MIB-переменной отображается последовательностью цифровых субполей, характеризующих переменную, например: переменная 1.3.6.1.2.1.5 (в символьном выражении iso.org.dod.internet.mgmt.mib.icmp) характеризуется последовательностью кодов 0x2B 0x06 0x01 0x02 0x01 0x05 0x00.

Начиная с января 1998 года, выпущен набор документов, посвященных **SNMPv3**. В этой версии существенно расширена функциональность (см. таблицу 1 тип PDU=5-8), разработана система безопасности.

В данной версии реализована модель, базирующаяся на процессоре SNMP (SNMP Engene) и содержащая несколько подсистем (дипетчер, система обработки сообщений, безопасности и управления доступом, см. рис. 4.4.13.4).

Перечисленные подсистемы служат основой функционирования генератора и обработчика команд, отправителя и обработчика уведомлений и прокси-сервера (Proxy Forwarder), работающих на прикладном уровне. Процессор SNMP идентифицируется с помощью snmpEngineID.

Обеспечение безопасности модели работы SNMP упрощается обычно тем, что обмен запросами-откликами осуществляется в локальной сети, а источники запросов-откликов легко идентифицируются.

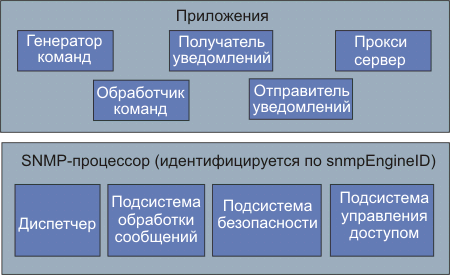


Рис. 5 - Архитектура сущности SNMP (SNMP-entity)

Компоненты процессора SNMP перечислены в таблице 4.4.13.5 (смотри RFC 2571 и -2573)

Таблица 7 - Компоненты процессора SNMP

|  |  |
| --- | --- |
| Название компонента | Функция компонента |
| Диспетчер | Позволяет одновременную поддержку нескольких версий SNMP-сообщений в процессоре SNMP. Этот компонент ответственен за прием протокольных блоков данных (PDU), за передачу PDU подсистеме обработки сообщений, за передачу и прием сетевых SNMP-сообщений |
| Подсистема обработки сообщений | Ответственна за подготовку сообщений для отправки и за извлечение данных из входных сообщений |
| Подсистема безопасности | Предоставляет услуги, обеспечивающие безопасность: аутентификацию и защищенность сообщений от перехвата и искажения. Допускается реализация нескольких моджелей безопасности |
| Подсистема управления доступом | Предоставляет ряд услуг авторизации, которые могут использоваться приложениями для проверки прав доступа. |
| Генератор команд | Инициирует SNMP-запросы Get, GetNext, GetBulk или Set, предназначенные для локальной системы, которые могут использоваться приложениями для проверки прав доступа. |
| Обработчик команд | Воспринимает SNMP-запросы Get, GetNext, GetBulk или Set, предназначенные для локальной системы, это индицируется тем, что contextEngeneID в полученном запросе равно соответствующему значению в процессоре SNMP. Приложение обработчика команд выполняет соответствующие протокольные операции, генерирует сообщения отклика и посылает их откправителю запроса. |
| Отправитель уведомлений | Мониторирует систему на предмет выявления определенных событий или условий и генерирует сообщения Trap или Inform. Источник уведомлений должен иметь механизм определения адресата таких сообщений, а также параметров безопасности |
| Получатель уведомлений | Прослушивает сообщения уведомления и формирует сообщения-отклики, когда приходит сообщение с PDU Inform |
| Прокси-сервер | Переадресует SNMP-сообщения. Реализация этогог модуля является опционной |

На рис. 6 показан формат сообщений SNMPv3, реализующий модель безопасности UBM (User-Based Security Model).

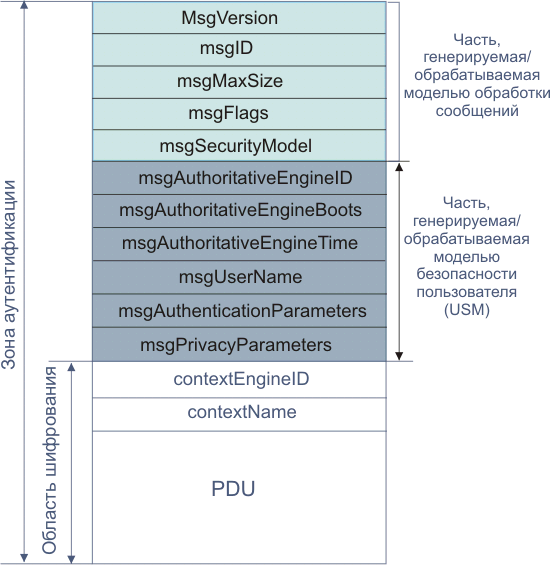


Рис.6 - Формат сообщений SNMPv3 c UBM

Первые пять полей формируются отправителем в рамках модели обработки сообщений и обрабатываются получателем. Следующие шесть полей несут в себе параметры безопасности. Далее следует PDU (блок поля данных) с **contextEngeneID** и **contextName**.

* *msgVersion* (для SNMPv3)=3
* *msgID* - уникальный идентификатор, используемый SNMP-сущностями для установления соответствия между запросом и откликом. Значение msgID лежит в диапазоне 0 - (231-1)
* *msgMaxSize* - определяет максимальный размер сообщения в октетах, поддерживаемый отправителем. Его значение лежит в диапазоне 484 - (231-1) и равно максимальному размеру сегмента, который может воспринять отправитель.
* *msgFlags* - 1-октетная строка, содержащая три флага в младших битах: **reportableFlag, privFlag, authFlag**. Если reportableFlag=1, должно быть прислано сообщение с PDU Report; когда флаг =0, Report посылать не следует. Флаг reportableFlag=1 устанавливается отправителем во всех сообщениях запроса (Get, Set) или Inform. Флаг устанавливается равным нулю в откликах, уведомлениях Trap или сообщениях Report. Флаги privFlag и authFlag устанавливаются отправителем для индикации уровня безопасности для данного сообщения. Для privFlag=1 используется шифрование, а для authFlag=0 - аутентификация. Допустимы любые комбинации значений флагов кроме privFlag=1 AND authFlag=0 (шифрование бех аутентификации).
* *msgSecurityModel* - идентификатор со значением в диапазоне 0 - (231-1), который указывает на модель безопасности, использованную при формировании данного сообщения. Зарезервированы значения 1 для SNMPv1,2 и 3 - для SNMPv3.

Модель безопасности **USM** (User-Based Security Model) использует концепцию авторизованного сервера (authoritative Engene). При любой передаче сообщения одна или две сущности, передатчик или приемник, рассматриваются в качестве авторизованного SNMP-сервера. Это делается согласно следующим правилам:

* Когда SNMP-сообщение содержит поле данных, которое предполагает отклик (например, Get, GetNext, GetBulk, Set или Inform), получатель такого сообщения считается авторизованным.
* Когда SNMP-сообщение содержит поле данных, которое не предполагает посылку отклика (например, SNMPv2-Trap, Response или Report), тогда отправитель такого сообщения считается авторизованным.

Таким образом, сообщения, посланные генератором команд, и сообщения Inform, посланные отправителем уведомлений, получатель является авторизованным. Для сообщений, посланных обработчиком команд или отправителем уведомлений Trap, отправитель является авторизованным. Такой подход имеет две цели:

* Своевременность сообщения определяется с учетом показания часов авторизованного сервера. Когда авторизованный сервер посылает сообщение (Trap, Response, Report), оно содержит текущее показание часов, так что неавторизованный получатель может синхронизовать свои часы. Когда неавторизованный сервер посылает сообщение (Get, GetNext, GetBulk, Set, Inform), он помещает туда текущую оценку показания часов места назначения, позволяя получателю оценить своевременность прихода сообщения.
* Процесс локализации ключа, описанный ниже, устанавливает единственного принципала, который может владеть ключем. Ключи могут храниться только в авторизованном сервере, исключая хранение нескольких копий ключа в разных местах.

Когда исходящее сообщение передается процессором сообщений в USM, USM заполняет поля параметров безопасности в заголовке сообщения. Когда входное сообщение передается обработчиком сообщений в USM, обрабатываются значения параметров безопасности, содержащихся в заголоке сообщения. В параметрах безопасности содержатся:

* *msgAuthoritativeEngeneID* - snmpEngeneID авторизованного сервера, участвующего в обмене. Таким образом, это значение идентификатора отправителя для Trap, Response или Report или адресата для Get, GetNext, GetBulk, Set или Inform.
* *msgAuthoritativeEngeneBoots* - snmpEngeneBoots авторизованного сервера, участвующего в обмене. Объект snmpEngeneBoots является целым в диапазоне 0 - (231-1). Этот код характеризует число раз, которое SNMP-сервер был перезагружен с момента конфигурирования.
* *msgAuthoritativeEngeneTime* - nmpEngeneTime авторизованного сервера, участвующего в обмене. Значение этого кода лежит в диапазоне 0 - (231-1). Этот код характеризует число секунд, которое прошло с момента последней перезагрузки. Каждый авторизованный сервер должен инкрементировать этот код один раз в секунду.
* *msgUserName* - идентификатор пользователя от имени которого послано сообщение.
* *msgAuthenticationParameters* - нуль, если при обмене не используется аутентификация. В противном случае - это аутентификационный параметр.
* *msgPrivacyParameters* - нуль - если не требуется соблюдения конфимденциальности. В противном случае - это параметр безопасности. В действующей модели USM используется алгоритм шифрования DES.

Механизм аутентификации в SNMPv3 предполагает, что полученное сообщение действительно послано принципалом, идентификатор которого содержится в заголовке сообщения, и он не был модифицирован по дороге. Для реализации аутентификации каждый из принципалов, участвующих в обмене должен иметь секретный ключ аутентификации, общий для всех участников (определяется на фазе конфигурации системы). В посылаемое сообщение отправитель должен включить код, который является функцией содержимого сообщения и секретного ключа. Одним из принципов USM является прверка своевременности сообщения (смотри выше), что делает маловероятной атаку с использованием копий сообщения.

Система конфигурирования агентов позволяет обеспечить разные уровни доступа к MIB для различных SNMP-менеджеров. Это делается путем ограничения доступа некоторым агантам к определенным частям MIB, а также с помощью ограничения перечня допустимых операций для заданной части MIB. Такая схема управления доступом называется **VACM** (View-Based Access Control Model). В процессе управления доступом анализируется контекст (vacmContextTable), а также специализированные таблицы vacmSecurityToGroupTable, vacmTreeFamilyTable и vacmAccessTable.

SNMP-протокол служит примером системы управления, где для достижения нужного результата выдается не команда, а осуществляется обмен информацией, решение же принимается "на месте" в соответствии с полученными данными. Внедрены подситемы аутентификации, информационной безопасности и управления доступом.

Структура SNMP MIB

Обрабатываемый агентом список объектов и их типов закладывается в него разработчиком, а станция управления получает эту информацию с помощью MIB (Management Information Base). MIB - текстовый файл, описывающий доступные объекты и их типы на языке, определяемом стандартом SMI (Structure and Identification of Management Information). Агент не использует этот файл при работе. MIB делится на модули, некоторые модули принимаются в виде стандартов, некоторые модули создаются разработчиками оборудования.

Разработчик управляемого оборудования (разработчик агента) должен предоставить список поддерживаемых агентом модулей. При описании модуля указывается какие объекты обязательны для реализации, а какие - нет. При описании агента можно указывать какие модули он поддерживает, в каком объеме и с какими модификациями.

Ha сегодня существует несколько стандартов на базы данных управляющей информации для протокола SNMP. Основными являются стандарты MIB-I и MIB-II, а также версия базы данных для удаленного управления RMON MIB. Кроме этого существуют стандарты для специальных устройств MIB конкретного типа (например, MIB для концентраторов или MIB для модемов), а также частные MIB конкретных фирм-производителей оборудования.

Первоначальная спецификация MIB-I определяла только операции чтения значений переменных. Операции изменения или установки значений объекта являются частью спецификаций MIB-II.

База данных MIB-II не дает детальной статистики по характерным ошибкам кадров Ethernet, кроме этого, она не отражает изменение характеристик во времени, что часто интересует сетевого администратора.

Эти ограничения были впоследствии сняты новым стандартом на MIB — RMON MIB, который специально ориентирован на сбор детальной статистики по протоколу Ethernet, к тому же с поддержкой такой важной функции, как построение агентом зависимостей статистических характеристик от времени.

**3.3 Недостатки протокола SNMP**

Протокол SNMP служит основой многих систем управления, хотя имеет несколько принципиальных недостатков, которые перечислены ниже.

* Отсутствие средств взаимной аутентификации агентов и менеджеров. Единственным средством, которое можно было бы отнести к средствам аутентификации, является использование в сообщениях так называемой «строки сообщества» — «community string». Эта строка передается по сети в открытой форме в сообщении SNMP и служит основой для деления агентов и менеджеров на «сообщества», так что агент взаимодействует только с теми менеджерами, которые указывают в поле community string ту же символьную строку, что и строка, хранящаяся в памяти агента. Это, безусловно, не способ аутентификации, а способ структурирования агентов и менеджеров. Версия SNMP v.2 должна была ликвидировать этот недостаток, но в результате разногласий между разработчиками стандарта новые средства аутентификации хотя и появились в этой версии, но как необязательные.
* Работа через ненадежный протокол UDP (а именно так работает подавляющее большинство реализаций агентов SNMP) приводит к потерям аварийных сообщений (сообщений trap) от агентов к менеджерам, что может привести к некачественному управлению. Исправление ситуации путем перехода на надежный транспортный протокол с установлением соединений чревато потерей связи с огромным количеством встроенных агентов SNMP, имеющихся в установленном в сетях оборудовании. (Протокол CMIP изначально работает поверх надежного транспорта стека OSI и этим недостатком не страдает.) Разработчики платформ управления стараются преодолеть эти недостатки. Например, в платформе HP OV Telecom DM TMN, являющейся платформой для разработки многоуровневых систем управления в соответствии со стандартами TMN и ISO, работает новая реализация SNMP, организующая надежный обмен сообщениями между агентами и менеджерами за счет самостоятельной организации повторных передач сообщений SNMP при их потерях.