1 Протоколы взаимодействия вакуумного оборудования

1.1 Модель OSI

Модель *OSI\ISO* – идеальная модель построения сети.

На рис. **1** показана структура идеальной модели *OSI*. Модель определяет различные уровни взаимодействия систем. Каждый уровень выполняет определённые функции при таком взаимодействии.

Данные	Прикладной доступ к сетевым службам
Данные	Представления представление и кодирование данных
Данные	Сеансовый Управление сеансом связи
Блоки	Транспортный безопасное и надёжное соединие точка-точка
Пакеты	Сетевой Определение пути и IP (логическая адресация)
Кадры	Канальный MAC и LLC (Физическая адресация)
Биты	Физический кабель, сигналы, бинарная передача данных

Рис. 1. Уровни модели OSI

Уровни модели OSI делятся на [1] :

— **физический**: предназначен непосредственно для передачи потока данных. Осуществляет передачу электрических или оптических сигналов в кабель или в радиоэфир и, соответственно, их приём и преобразование в

биты данных в соответствии с методами кодирования цифровых сигналов;

— **канальный**: предназначен для обеспечения взаимодействия сетей на физическом уровне. Полученные с физического уровня данные проверяет на ошибки, если нужно исправляет, упаковывает во фреймы, проверяет на целостность, и отправляет на сетевой уровень;

— **сетевой**: определяет пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, за определение кратчайших маршрутов, коммутацию и маршрутизацию, за отслеживание неполадок и заторов в сети;

— **транспортный**: организует доставку данных без ошибок, потерь и дублирования (не все) в той последовательности, как они были переданы. Разделяет данные на фрагменты равной величины, объединяя короткие и разбивая длинные (размер фрагмента зависит от используемого протокола);

— **сеансовый**: управляет созданием/завершением сеанса связи, обменом информацией, синхронизацией задач, определением права на передачу данных и поддержанием сеанса в периоды неактивности приложений. Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия;

— **представления**: на этом уровне может осуществляться преобразование протоколов и сжатие/распаковка или кодирование/декодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально;

— **прикладной**: уровень приложений (англ. Application layer). Обеспечива-

ет взаимодействие сети и приложений пользователя, выходящих за рамки модели OSI. На этом уровне работают изученные в работе протоколы автоматизации промышленных сетей.

1.2 Протокол Modbus

В настоящее время существует огромное множество промышленных протоколов, используемых в автоматизации. На рис. **2** показано распределение рынка между протоколами [2]:

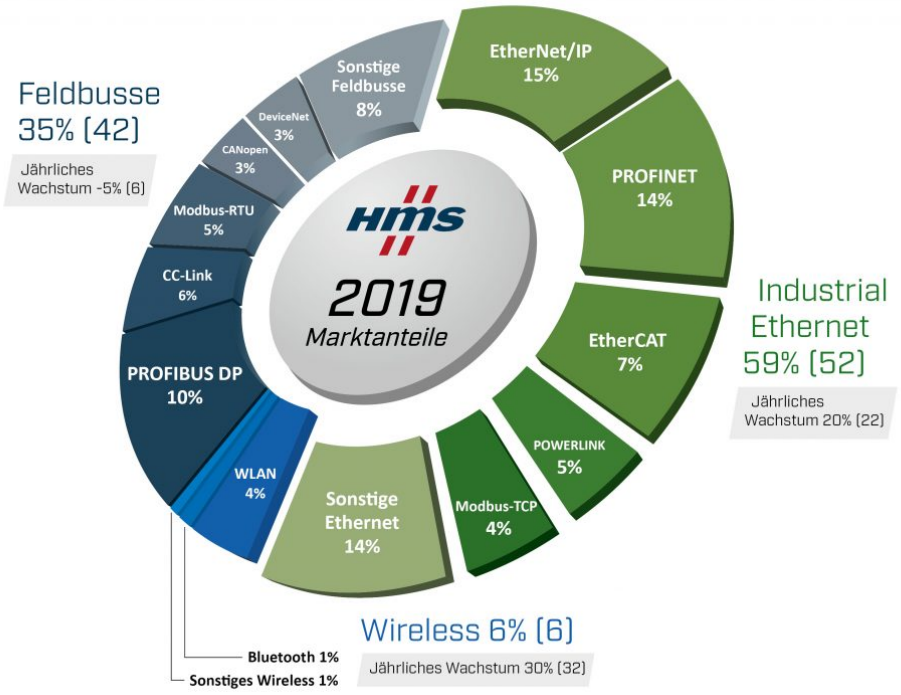

Рис. 2. Диаграмма долей рынка промышленных протоколов

В России основную долю рынка занимают протоколы *Modbus* и *Profibus* [3], однако *Profibus* уступает своему конкуренту в ряде вещей:

— стандарт открыт не полностью;

— сложнее в освоении, чем *Modbus*;

— в СНГ *Modbus* более популярен.

1.3 Прицип работы протокола Modbus

1.3.1 Физический уровень

Протокол *Modbus* может быть использован со следующими интерфейсами:

— **RS-232/422/485**: последовательные интерфейсы, широко распространенные в промышленности. Интерфейсы RS-422/485 обеспечивают дальность сигнала до 1200 метров. Используются протоколы *Modbus RTU/ASCII*

— **TCP/IP**: физическим каналом передачи данных могут любые ethernet-интерфейсы. Используется протокол *Modbus TCP*.

Существует 3 разновидности протокола *Modbus* [4]:

1. *Modbus ASCII*: в котором данные кодируются символами из таблицы ASCII (рис. 1) и передаются в шестнадцатеричном формате и данный формат протокола встречается довольно редко;

2. *Modbus RTU*: самый распространенный вариант протокола Modbus, который кодирует данные в двоичном формате и разделяет пакеты с помощью временного интервала;

3. *Modbus TCP*: данные кодируются в двоичном формате и упаковываются в TCP - пакет, для передачи по IP-сетям и предназначен для работы в локальных сетях.

На рис. **3** показан пример построения схемы контроля за неким объектом при помощи протокола *Modbus* [5].

1.3.2 Логический уровень

Протокол *Modbus* предполагает одно ведущее устройство и до 247 ведомых. Обмен данными начинается ведущим устройством. Ведомые не могут на-

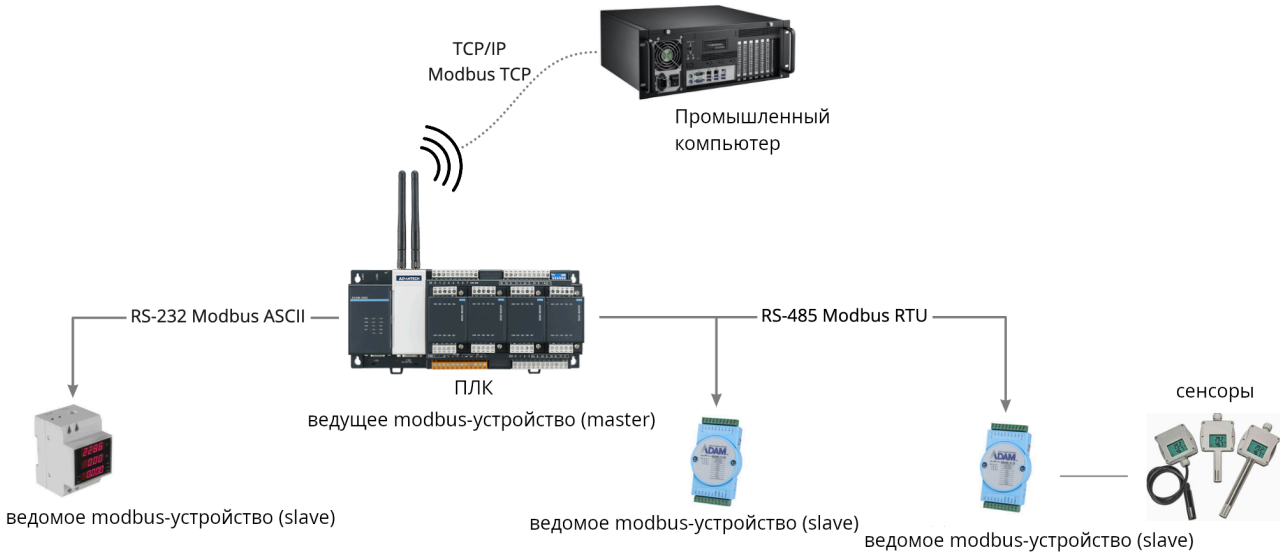

Рис. 3. Физический уровень протокола *Modbus*

чинать передачу и обмениваться данными между собой. В любой момент времени может происходить только один акт обмена. Структуры пакетов *Modbus* при работе 3 способами приведены на рис. **4**.

1.3.2.1 Modbus RTU

Сообщение начинает восприниматься как новое после паузы длиной в 14 бит. На рис. **5** показан формат пакета. У пакета есть следующие поля [3]:

— **Адрес:** содержит адрес ведомого устройства. Адрес отправляется даже при ответе на запрос мастера, тем самым всегда понятно откуда пришёл ответ;

— **Код функции:** говорит модулю о том, что ему необходимо сделать;

— **Данные:** тут может содержаться информация о параметрах, которые используются в исполнении команд мастера или показания, передаваемые мастеру;

— **Контрольная сумма:** используется для проверки целостности пакета.

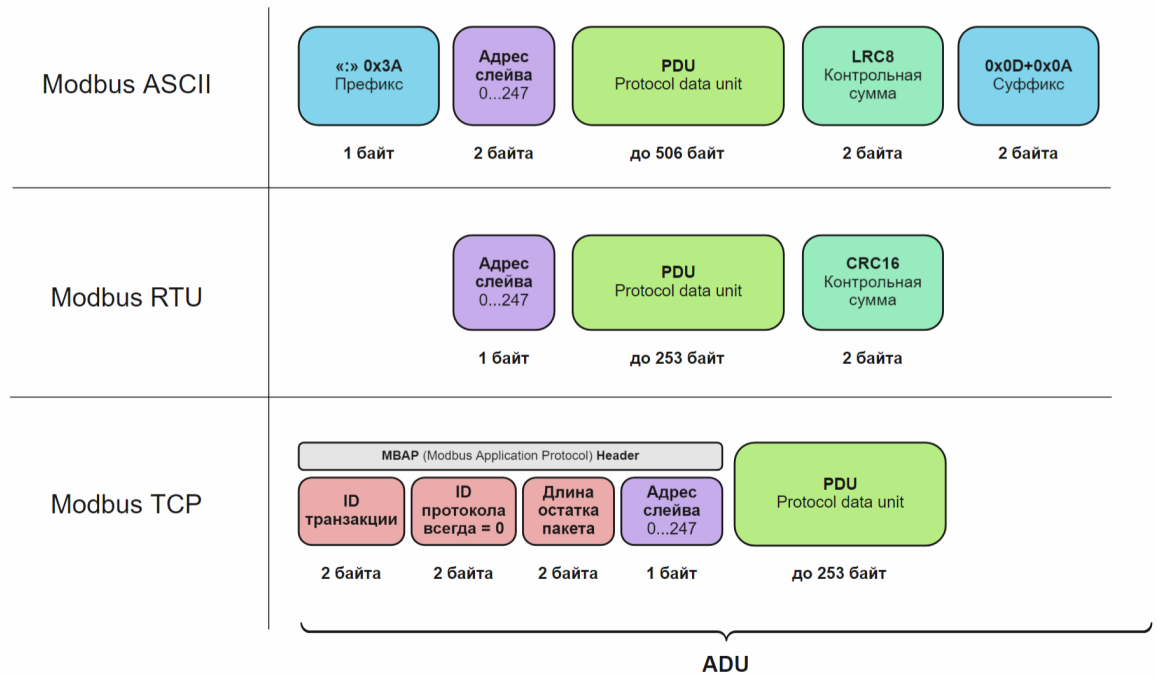

Рис. 4. Структура пакета *Modbus*

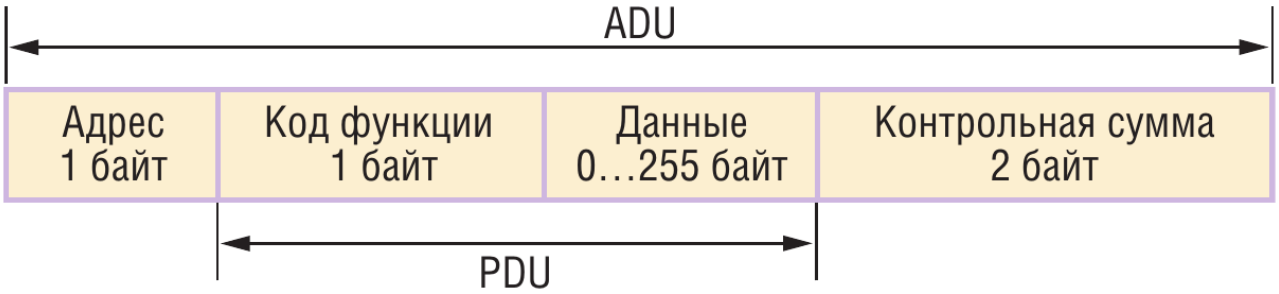

Рис. 5. Формат кадра протокола *Modbus RTU*

1.3.2.2 Modbus TCP

Данный протокол используется для того, чтобы подключить устройства, работающие по протоколу *Modbus* к сети *Internet* [6]. То есть, в соответствии со стандартом *OSI\ISO* (см. рис. **1**) на транспортном уровне используется протокол *TCP*, а на прикладном – *Modbus*. В этом случае проверка целостности пакета ложится на протокол *TCP*. Структура протокола *Modbus TCP* приведена на рис. **6**. У пакета есть следующие поля [3]:

— **Идентификатор обмена:** используется для идентификации сообщения в случае, когда в пределах одного TCP - соединения клиент посылает серве-

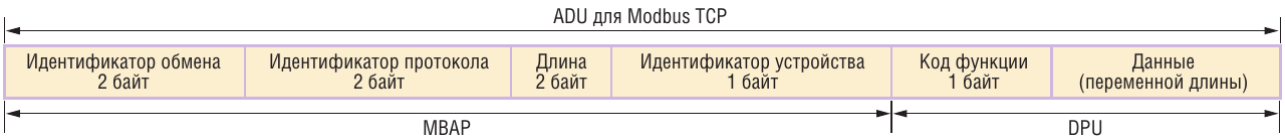

Рис. 6. Формат кадра протокола *Modbus TCP*

ру несколько сообщений без ожидания ответа после каждого сообщения;

— **Идентификатор протокола:** всегда выставлен на 0 (как и протокола *TCP*);

— **Длина:** указывает количество следующих байтов;

— **Идентификатор устройства:** адрес slave - устройства;

— **Код функции:** аналогично **1.3.2.1. Modbus RTU**;

— **Данные:** аналогично **1.3.2.1. Modbus RTU**.

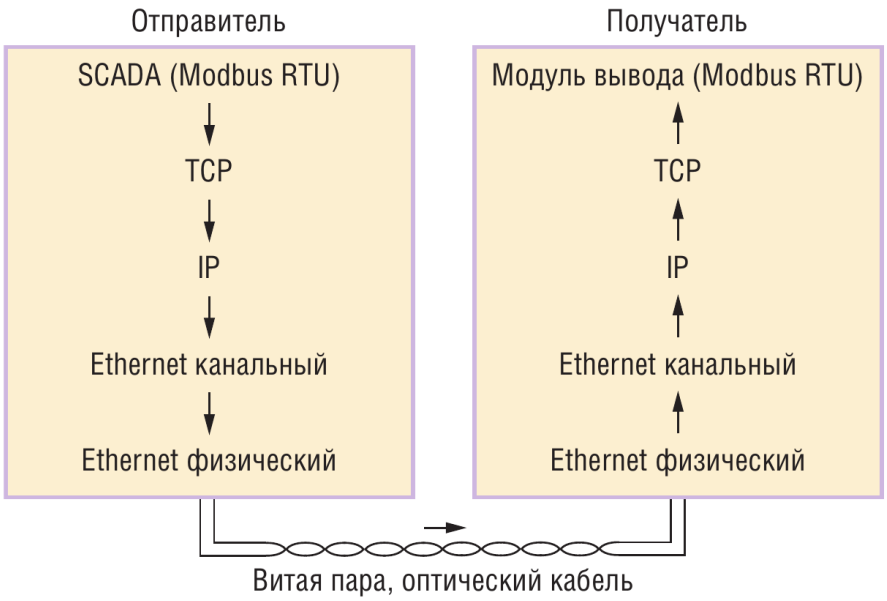

Рис. 7. Процесс передачи пакетов *Modbus RTU* по сети *TCP*

Как можно заметить, *Modbus RTU*, оказывается “вшит” в пакет *TCP*, тем самым получается *Modbus TCP*. На рис. **7** показан принцип работы такой системы [3]:

— коды функций передаются с прикладного уровня на транспортный (*Modbus–TCP*), добавление заголовка *TCP*;

— передача на сетевой уровень, добавление блока *IP*;

— передача на канальный уровень, а затем на физический (*Ethernet*)

После прохождения через канал связи пакет начинает обратное движение согласно модели *OSI\ISO* (см. рис. **1**).

Более подробно о работе с протоколом *Modbus* можно узнать в документации [7].