

## Оглавление

1	Структура сетевого ПО ОС Windows .....	1
1.1	Сетевой стек ОС Windows.....	1
1.2	Эталонная модель OSI.....	1
1.3	Сетевые компоненты Windows и их соответствие уровням OSI.....	2

## 1 Структура сетевого ПО ОС Windows

### 1.1 Сетевой стек ОС Windows

В операционную систему включена поддержка сетей, интегрированная с подсистемой **ввода-вывода и Win32 API**.

К четырем базовым типам сетевого программного обеспечения относятся:

- **Сетевые сервисы,**
- **Сетевые API,**
- **Сетевые протоколы**
- **Драйверы сетевых адаптеров.**

Все они располагаются один над другим, образуя сетевой стек.

Для каждого уровня в Windows предусмотрены четко определенные интерфейсы, поэтому в дополнение к набору API-функций, протоколов и драйверов адаптеров, поставляемых с Windows, сторонние разработчики могут создавать собственные компоненты, расширяющие сетевую функциональность операционной системы.

### 1.2 Эталонная модель OSI

Для стандартизации и интеграции сетевого программного обеспечения, международная организация по стандартизации (ISO) определила программную модель пересылки сообщений между компьютерами.

Эта модель получила название **эталонной модели OSI (Open Systems Interconnection)**. В ней определено семь уровней программного обеспечения.

Эталонная модель OSI — идеал, точно реализованный лишь в очень немногих системах, но часто используемый при объяснении основных принципов работы сети.

Каждый уровень на одном из узлов считает, что он взаимодействует с тем же уровнем на другом узле. На данном уровне оба узла «разговаривают» на одном языке, или протоколе.

В действительности сетевой запрос должен сначала пройти до самого нижнего уровня на первом узле, затем он передается по несущей среде и уже на втором узле вновь поднимается до уровня, который его поймет и обработает.

### **1.3 Сетевые компоненты Windows и их соответствие уровням OSI**

Между уровнями OSI и реальными сетевыми компонентами нет точного соответствия. Некоторые компоненты охватывают несколько уровней.

Задача каждого уровня состоит в том, чтобы предоставлять сервисы более высоким уровням и скрывать от них конкретную реализацию этих сервисов.

#### **7. Прикладной уровень (application layer).**

Обеспечивает передачу данных между двумя сетевыми приложениями, включая проверку прав доступа, идентификацию взаимодействующих машин и инициацию обмена данными.

#### **Сетевые приложения Windows.**

#### **6. Презентационный уровень (presentation layer).**

Отвечает за форматирование данных, в том числе решает вопросы сжатия данных, кодирования и т. д.

**Сетевые API(DLL сетевых API).** Обеспечивают независимое от протоколов взаимодействие приложений через сеть. Сетевые API реализуются либо в режиме ядра и пользовательском режиме, либо только в пользовательском режиме.

Некоторые сетевые API являются оболочками других API и реализуют специфическую модель программирования или предоставляют дополнительные сервисы (Термином «сетевые API» обозначаются любые программные интерфейсы, предоставляемые сетевым программным обеспечением.)

#### **5. Сеансовый уровень (session layer).**

Управляет соединением взаимодействующих приложений, включая высокоуровневую синхронизацию и контроль за тем, какое из них активное - «говорит», а какое пассивное - «слушает».

**Драйверы устройств режима ядра (Драйвер сетевого API – клиент TDI (Transport Driver Interface))**, обычно реализующие ту часть сетевого API, которая работает в режиме ядра.

Клиенты TDI называются так из-за того, что пакеты запросов ввода-вывода (IRP), которые они посылают драйверам протоколов, форматируются по стандарту Transport Driver Interface. Этот стандарт определяет общий интерфейс программирования драйверов устройств режима ядра.

#### **4. Транспортный уровень (transport layer).**

На передающей стороне разбивает сообщения на пакеты и присваивает им порядковые номера, гарантирующие прием пакетов в должном порядке. Кроме того, изолирует сеансовый уровень от влияния изменений в составе оборудования.

#### **3. Сетевой уровень (network layer).**

Создает заголовки пакетов, отвечает за маршрутизацию, контроль трафика и взаимодействие с межсетевой средой. Это самый высокий из уровней, который понимает топологию сетей, т. е. физическую конфигурацию машин в них, ограничения пропускной способности этих сетей и т. д.

**3-4 Драйверы протоколов режима ядра. (Драйвер протокола – транспорт TDI)**. Представляют собой драйверы протоколов режима ядра и часто называются транспортом, NDIS-драйверами протоколов или драйверами протоколов.

Они принимают пакеты запросов ввода-вывода (IRP) от клиентов TDI и обрабатывают запросы, представленные этими IRP.

#### **2. Канальный уровень (data-link layer).**

Пересылает низкоуровневые кадры данных, ждет подтверждений об их приеме и повторяет передачу кадров, потерянных в ненадежных линиях связи.

## Драйверы сетевых адаптеров (библиотека драйверов NDIS(Network Driver Interface Specification))

**Библиотека NDIS (Ndis.sys).** Драйверы сетевых адаптеров, соответствующие стандарту **NDIS (Network Driver Interface Specification)**. Инкапсулирует функциональность для драйверов адаптеров, скрывая от них специфику среды Windows , работающей в режиме ядра. Библиотека NDIS экспортирует функции для транспортов TDI, а также функции поддержки для драйверов адаптеров.

**Минипорт-драйверы NDIS.** Драйверы режима ядра, отвечающие за организацию интерфейсов между транспортом TDI и конкретными сетевыми адаптерами. Минипорт-драйверы NDIS пишутся так, чтобы они были заключены в оболочку библиотеки NDIS. Такая инкапсуляция обеспечивает межплатформенную совместимость с потребительскими версиями Microsoft Windows. Минипорт-драйверы NDIS взаимодействуют с сетевыми адаптерами, используя функции библиотеки NDIS, которые вызывают соответствующие функции HAL

**1. Физический уровень (physical layer).** Передает биты по сетевому кабелю или другой физической несущей среде. **Ethernet, IrDA, ATM, ...**