Сетевая подсистема Windows глазами разработчика.

Краткий, неполный и, в основном, неверный обзор. :-)

Алексей Пахунов

alexeypa@microsoft.com

Российские интернет-технологии

2011

Обо мне.

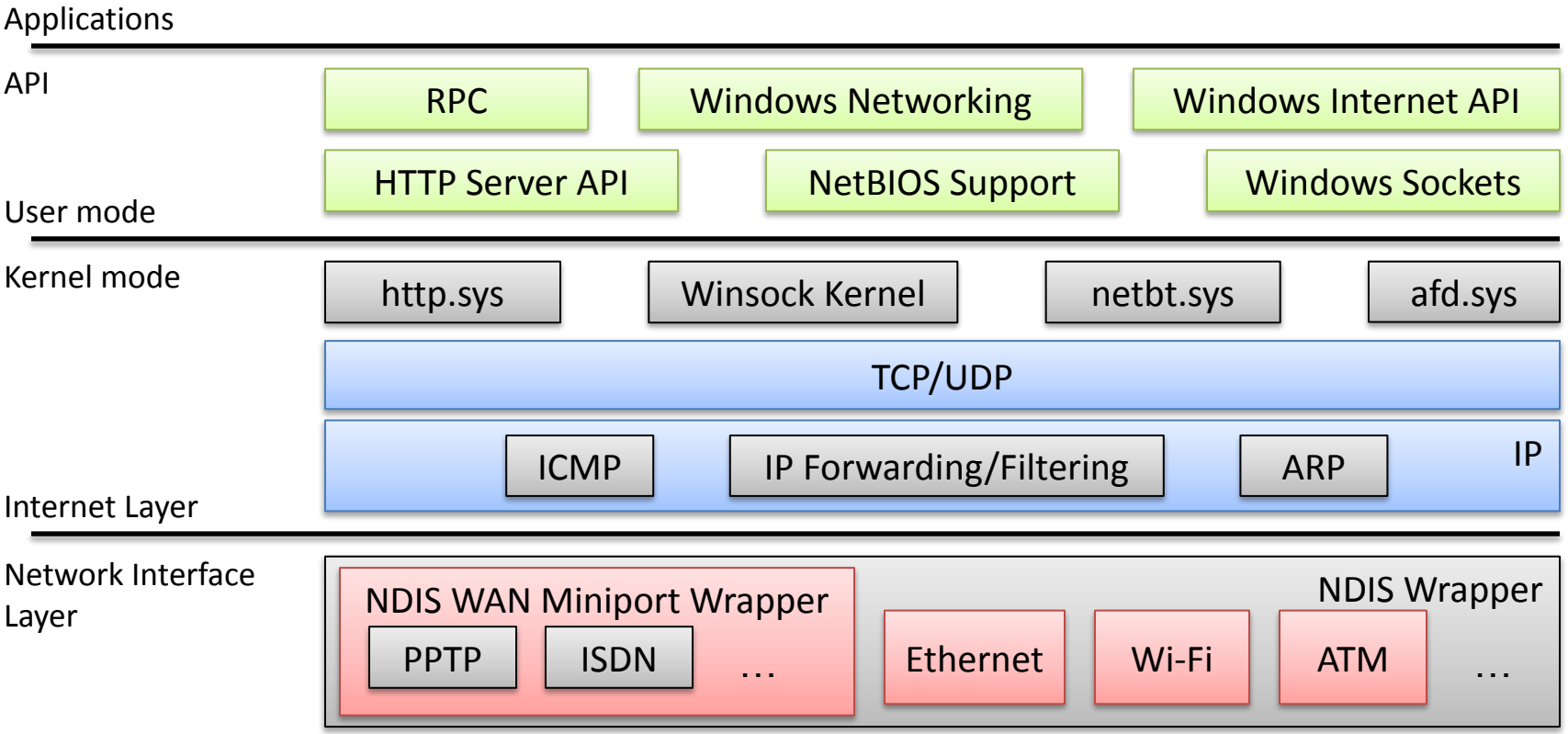
- Senior SDE в команде eXtreme Computing Group (XCG), Microsoft Research.
- Специализация: низкоуровневое и системное программирование; разработка драйверов и компонентов ядра Windows.
- 3 года в команде Windows Kernel: Wow64 и поддержка AVX.
- Мой блог: http://blog.not-a-kernel-guy.com.

Содержание.

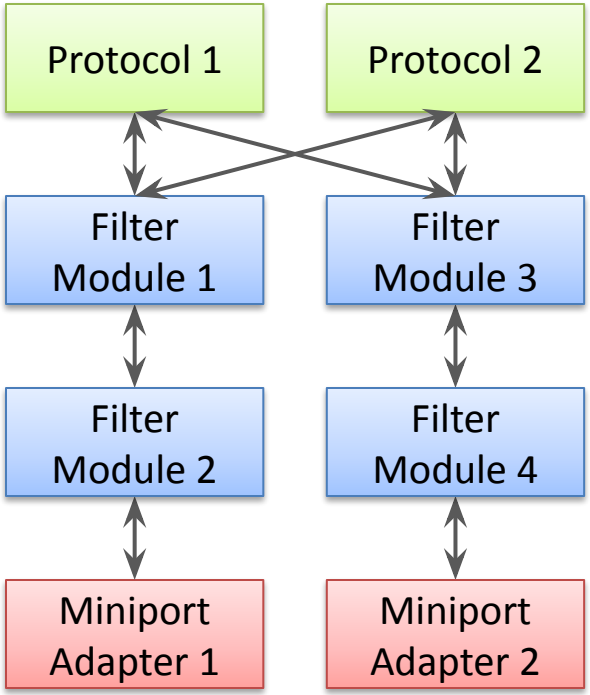
1. Архитектура стека TCP/IP.
2. Путь данных вверх и вниз.
3. Настройки и аппаратное ускорение.
4. Фильтры и мониторинг трафика.

АРХИТЕКТУРА СТЕКА TCP/IP.

Архитектура стека TCP/IP.

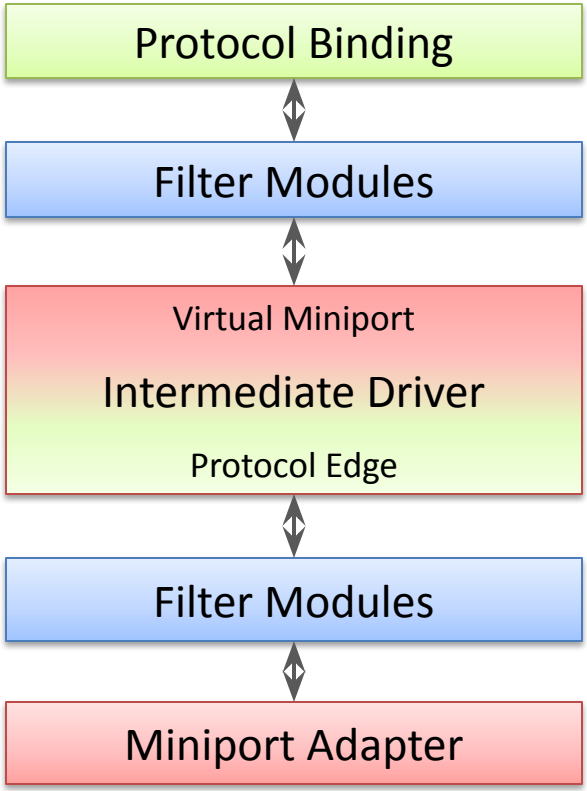


Стек драйверов в NDIS 6.0.



- Отдельный стек над каждым сетевым адаптером.
  - Многопортовые сетевые адаптеры могут запросить отдельный стек для каждого порта.
- Сетевой адаптер может привязывается к нескольким протоколам.
- Фильтры устанавливаются отдельно над каждым сетевым адаптером.

Промежуточные драйверы.



- Промежуточный драйвер объединяет два стека в один.
  - Верхний стек видит виртуальный сетевой адаптер.
  - Нижний стек привязывается к промежуточному драйверу как к протоколу.

Программные интерфейсы.

- Winsock (`send/recv, WSASend/WSARecv`).
- Winsock Kernel (`WskSend/WskReceive`).
- IP Helper.
- RPC (`RpcXxx`).
- WNet (`WNetXxx`).
- WinInet (`InternetXxx`).
- WinHTTP (`WinHttpXxx`).
- HTTP Server API (HttpXXX).

ПУТЬ ДАННЫХ ВВЕРХ И ВНИЗ.

Обработка принятых пакетов (IP).

- Сетевой адаптер проверяет целостность пакета и генерирует прерывание.
- Драйвер адаптера передает его выше по стеку.
- IP проверяет целостность IP заголовка, восстанавливает пакет из фрагментов, перенаправляет пакет согласно таблице маршрутизации.
- TCP/UDP проверяет целостность данных пакета, запрашивает повторную передачу и копирует данные в буфер приложения или драйвера:

```
recv(connection, buffer, length, 0);
```

Передача данных (TCP).

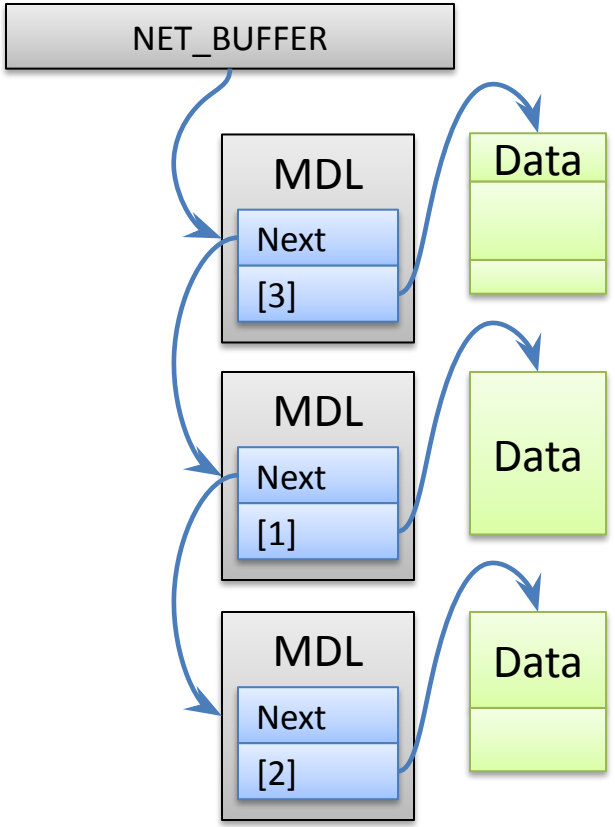
- Приложение указывает на данные для передачи:

```
send(connection, buffer, length, 0);
```

- TCP формирует заголовки пакета (или нескольких пакетов).
- IP формирует свои заголовки и разбивает пакеты на фрагменты, если необходимо.
- Драйвер адаптера ставит пакеты в очередь, настраивает DMA и запускает передачу пакетов.
- Сетевой адаптер генерирует прерывание по окончанию передачи.
- Драйвер адаптера возвращает буферы их владельцу.

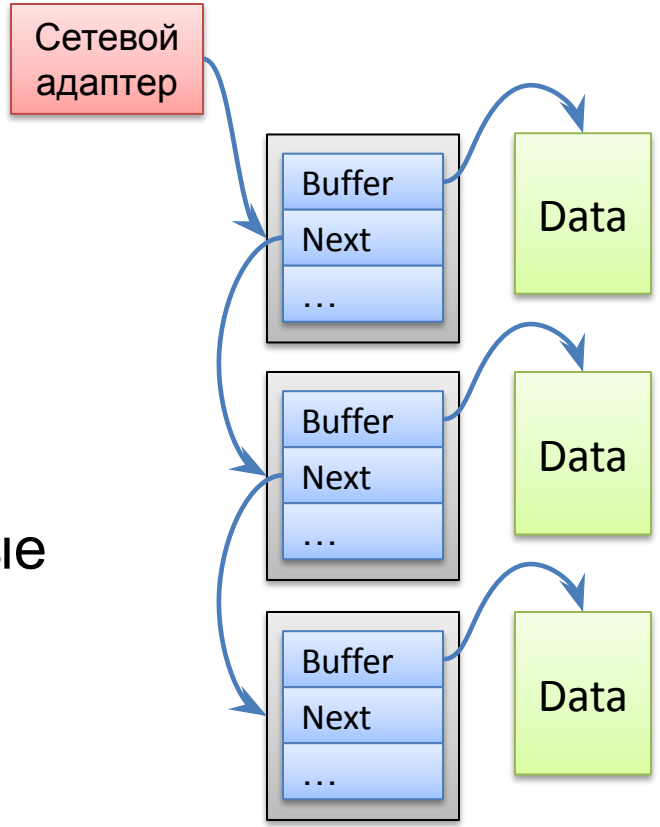
Как пакеты хранятся в памяти?

- Каждый пакет описывается списком буферов (NET_BUFFER).
  - Буфер может располагаться в несмежных физических страницах.
- Между уровнями передаются указатели.
  - Данные пакета копируются только один раз.



Прямой доступ в память (DMA).

- Сетевой адаптер поддерживает очереди буферов.
  - Несколько очередей для приёма и передачи.
- Драйвер отвечает за выделение памяти, вставляет буферы в очередь и удаляет их оттуда.
- Сетевой адаптер сохраняет принятые данные в подготовленные драйвером буфера.
- Дескрипторы указывают сетевому адаптеру как нужно «склеивать» пакеты из нескольких буферов.



Прерывания в Windows.

- Уровни прерываний (IRQL):
  - PASSIVE_LEVEL – обычный код; используются приоритеты потоков.
  - DISPATCH_LEVEL – планировщик потоков и подкачка страниц приостановлены.
  - DIRQLs – прерывания от менее приоритетных устройств заблокированы.
- Прерывание обрабатывается в два этапа:
  - Обработчик прерывания должен выполнить минимум работы максимально быстро.
  - Отложенный обработчик (DPC) выполняет оставшуюся работу.
- IRQL нельзя произвольно понижать.
- Каждое из ядер может находится на своем уровне прерываний.

Прерывания в NDIS.

- Основные прерывания: пакет принят и передан.
- Обработка принятых пакетов проходит на DISPATCH_LEVEL.
  - Любой драйвер в стеке имеет право передать обработку в рабочий поток (PASSIVE_LEVEL).
- Исходящие пакеты формируются на PASSIVE_LEVEL.
  - Любой драйвер в стеке имеет право повысить IRQL до DISPATCH_LEVEL.

Уведомление приложений.

- Все операции ввода-вывода асинхронны.
  - Синхронные `send()` и `recv()` эмулируются.
- Уведомление об окончании операции доставляется одним из стандартных способов:
  - APC, установка события, IO completion port, threadpool, опрос OVERLAPPED.
  - Драйверы, работающие через Winsock Kernel, используют IRP (I/O Request Packet).

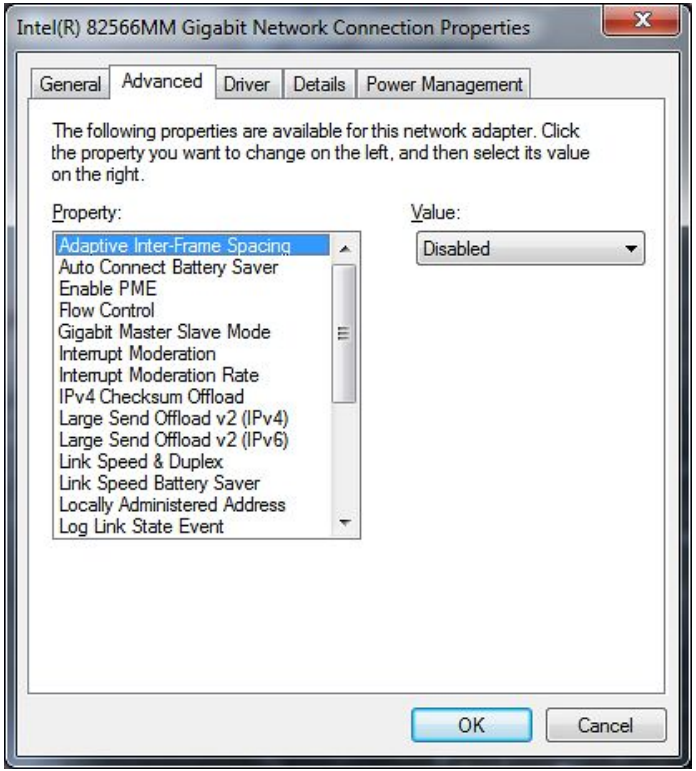
НАСТРОЙКИ И АППАРАТНОЕ УСКОРЕНИЕ.

Аппаратное ускорение.

- MAC и VLAN фильтры на сетевом адаптере.
- Регулирование частоты прерываний (Interrupt Moderation).
- Выгрузка вычислений на сетевой адаптер:
  - Вычисление и проверка контрольных сумм (Checksum Offloading).
  - TCP сегментация (Large Send Offloading).
  - TCP Chimney Offloading.
  - Обработка принятых пакетов на нескольких процессорах (Receive-Side Scaling).
- Поддержка виртуализации.

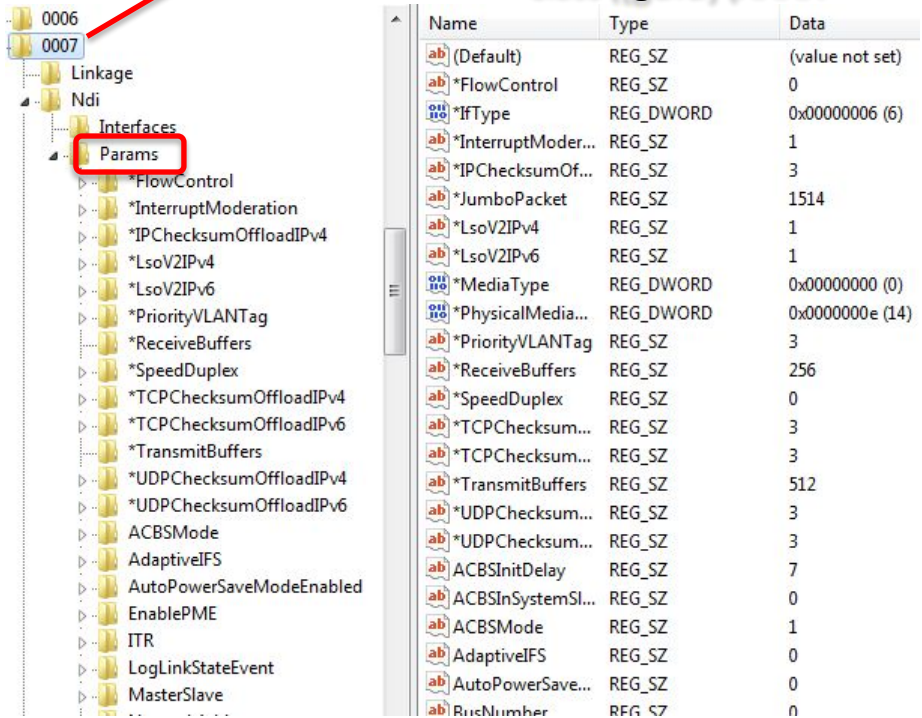
Настройка сетевого адаптера (1).

- Вкладка «Advanced».
  - Описывается в .INF файле драйвера.
- NDIS определяет стандартные параметры.
  - …но отображаемые названия параметров все равно берутся из .INF файла.



Настройка сетевого адаптера (2).

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{guid}\XXXX



Настройка TCP/IP.

- Доступные через реестр параметры TCP/IP описаны в TechNet и множестве других источников.
- HKLM\SYSTEM\CurrentControlSet\services\Tcpip\Parameters:
  - Адреса.
  - Размер окна TCP.
  - Маршрутизация.
  - Лимиты.
  - …

ФИЛЬТРЫ И СЛЕЖЕНИЕ ЗА ТРАФИКОМ.

NDIS фильтры.

- Делятся на следящие и модифицирующие фильтры.
- Перехватывают и пакеты, и управляющие OID запросы.
  - Иными словами – полностью контролируют нижнюю часть стека.
- Загружаются для всех адаптеров данного типа.
  - Перехватываемые функции конфигурируются для отдельно для каждого адаптера.

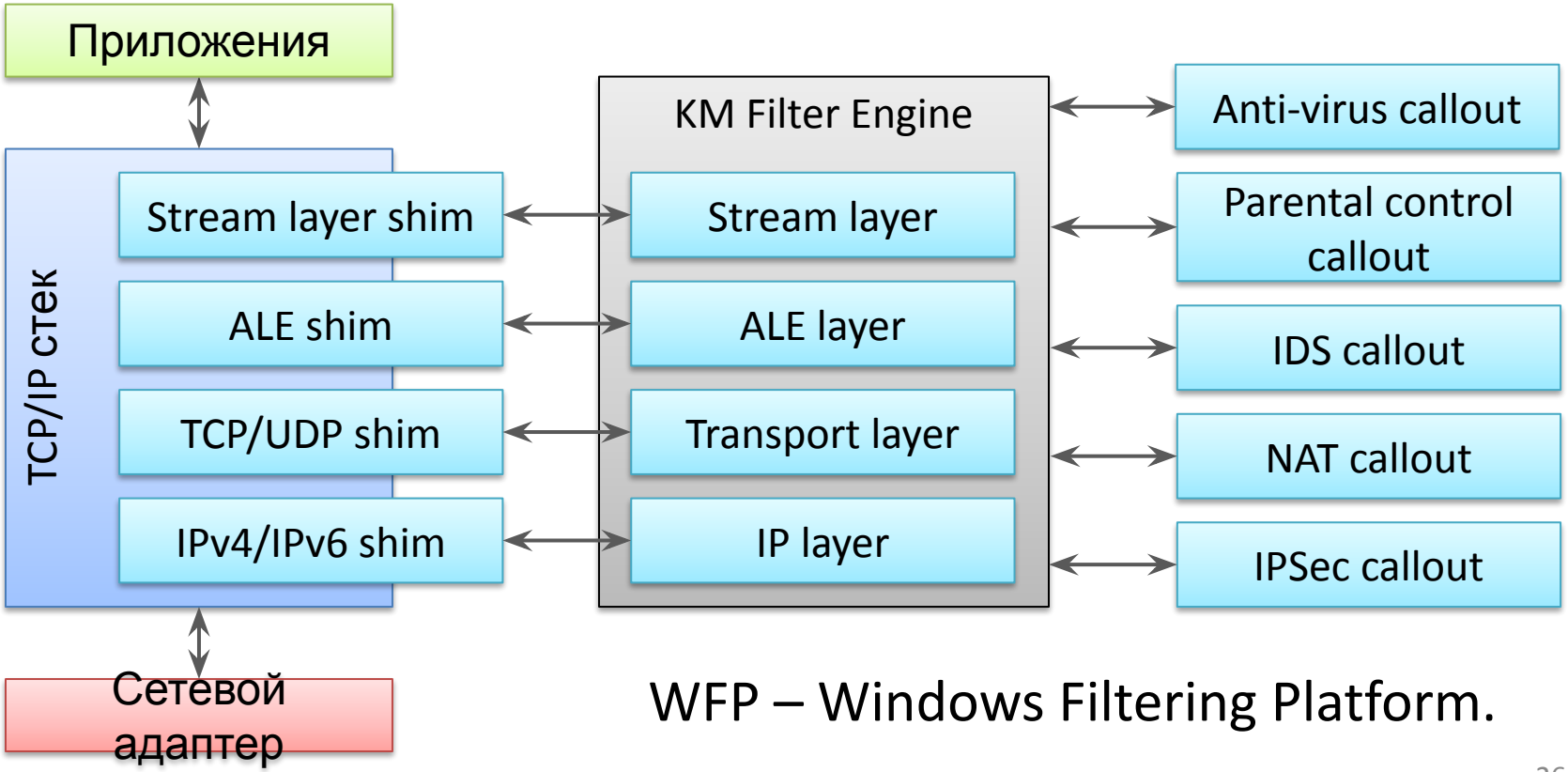
Расширение отладчика !ndiskd.

- Расширение !ndiskd:
  - Входит в состав Windows Debugging Tools.
  - Дружественно к неподготовленному пользователю.
  - Показывает детальную информацию об адаптерах, фильтрах и протоколах.
- Требует подключения ядерного отладчика.
  - Достаточно локального подключения.

Это проще, чем кажеться...

- Приостановите BitLocker.
- “`bcdedit /debug on`”.
- После перезагрузки: “`windbg.exe -kl`”.
- Убедитесь в корректности “`.sympath`”.
- “`!ndiskd.help`”.

Архитектура WFP.

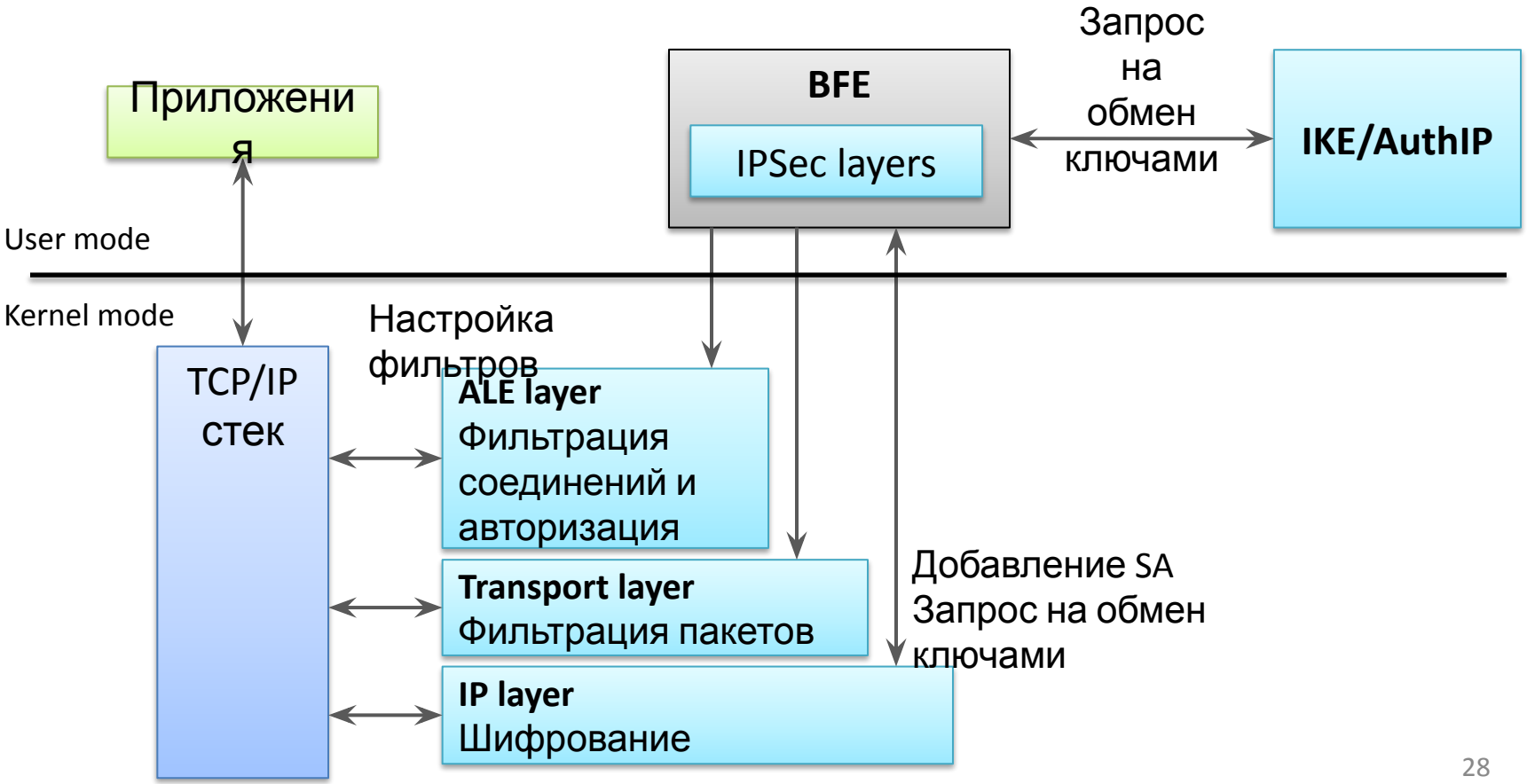


WFP – Windows Filtering Platform.

Основные элементы WFP.

- Shims:
  - Стек TCP/IP определяет несколько ключевых точек, где происходит фильтрация трафика.
- Filters:
  - Ко входящему и исходящему трафику применяется набор правил, задающий действия, применяемые к данным.
- Layers:
  - Фильтры групприрутся по уровням и подуровням.
  - Каждый уровень определяет свой набор полей для фильтрации.
  - Порядок применения фильтров однозначно определён.
- Callouts:
  - Фильтр может принять решение о глубокой инспекции пакета.

Архитектура IPsec.



Мониторинг WFP.

- Аудит:
  - Конфигурация WFP.
  - Отброшенные/пропущенные пакеты, соединения, операции с сокетами.
  - Обмен ключами и отброшенные пакеты в IPsec.
- Конфигурация WFP доступна через Win32 API.