

Часть 1.

Глава 1.Сетевые модели TCP/IP и OSI

Сетевая модель - полное описание (набор документации), достаточное для создание полностью функционирующей сети.

На данный момент часто упоминается 2 сетевые модели, которые разделены на уровни. Многоуровневость моделей упрощает ее изучение, упрощает разработку устройств (есть описание, какого формата данные на каком уровне получаются и отдаются), упрощает разработку ПО (например, при разработке браузера не нужно беспокоиться об уровнях 1-4)

Модель OSI и TCP\IP

Таблица 1.1.1 - Уровни моделей OSI и TCP\IP

OSI	TCP\IP (5 уровней, в 4х уровневой модели 1й и 2й уровень объединены в канальный)
7. Приложений	5-7 Приложений
6. Представлений	
5. Сеансовый	
4. Транспортный	4. Транспортный
3. Сетевой	3. Сетевой
2. Канальный	2. Канальный
1. Физический	1. Физический

Исторически сложилось, что модель TCP\IP более распространена в применении, но терминология OSI используется везде.

Важно понимать, что функции схожи, но начинка уровня различается в разных моделях, например, на уровне 3 в TCP\IP может быть ipv4 или ipv6, а в OSI какой-либо иной протокол, так как протокол может отличаться от IP.

Таблица 1.1.2 - Функционал уровней моделей OSI и TCP/IP

	OSI	TCP/IP
7	Интерфейс между приложениями и коммуникационным ПО	Интерфейс между ПО и сетью (стандартизирует службы, используемые приложениями, например, протокол http)
6	Преобразование данных в различные форматы (например, в какую-то кодировку текста или в бинарный файл)	
5	Управление сеансами	
4	Разбиение данных на сегменты и сборка сегментов на конечном устройстве, решение проблем при доставке данных до конечного устройства (коррекция ошибок при передаче, контроль доставки)	Аналогично
3	Соединение конечных устройств сети, состоит из: - Логическая адресация (определяет формат адреса, как используется адрес); - Маршрутизация (перенаправление данных до адреса); Определение маршрутов в сети (определение маршрутов до адреса и выбор оптимального)	Аналогично, но сеть = интернет
2	Обеспечение передачи данных по физическому каналу (как именно пересылать данные в конкретной среде, формат заголовков и окончаний пакетов второго уровня)	Аналогично
1	Описание физической среды передачи (расположение выводов, их назначение, уровни напряжений, кодирование сигнала в среде и тд.)	Аналогично

Взаимодействие разных уровней модели, инкапсуляция.

Каждый уровень имеет свою функцию, свой протокол. Для того, чтобы открыть в браузере страницу сайта, часто будут взаимодействовать все уровни. На примере TCP/IP модели (упрощенно, упущены детали, например, tls шифрование и трехэтапное рукопожатие для TCP):

0. (не входит в модель) Запрос в dns, чтобы получить ip адрес, который отвечает за интересующий сайт.

1. Уровень приложений, https протокол, создается запрос (сообщение, которое сообщает, что браузер хочет получить страницу) в формате https.

2. Запрос нужно доставить, поэтому происходит обращение к следующему уровню, транспортному. В данном примере на транспортном уровне используется TCP. К

данным уровня приложений добавляется заголовок TCP (инкапсуляция).

3. Транспортный уровень обращается к сетевому уровню для перенаправления пакетов до адресата. К данным транспортного уровня добавляется заголовок IP протокола (инкапсуляция, итоговые данные IP{TCP{HTTPS}})

4. Транспортный уровень обращается к уровню канала, данные инкапсулируются в очередной протокол, например, Ethernet.

5. Остается передача данных на физическом уровне, передача битов на ближайший маршрутизатор.

6. На маршрутизаторе анализируется заголовок IP протокола для определения, куда отправить фрейм дальше.

7. После прохождения цепочки маршрутизаторов, пакет попадает к адресату.

8. Получатель получает фрейм через физическую среду и декапсулирует данные в обратном порядке, в соответствии с протоколом (например, проверяется целостность сообщения). Из фрейма получает (с помощью декапсуляции) пакет IP (сетевого уровня), и т.д, поднимаясь до данных уровня приложения.

9. Получатель (сервер) создает данные формата https (заголовок + страница) и, проходя такие же стадии, как и отправитель, отправляет свой ответ.