**Case 2:**

**DNS:** В современном мире, большинство пользователей интернета видят сайты как Astafyev.ru, за каждым названием сайта стоит адрес своего сервера. Когда мы вбиваем в браузере название сайта, то наш запрос уходит к провайдеру услуг связи. Исторически на заре интернета был просто файл HOSTS.txt где каждый компьютер проверял какое имя соответствует IP адресу (Internet Protocol). Браузер перед тем как пойти к провайдеру и спрсить у него, проверяет свой кэш DNS. Далее провайдер обращается к resolver DNS - Domain Name System (Фактически данный сервер похож чем то на записную книжку в телефоне – есть имя фамилия, а под ней телефон. Так удобнее для пользователя хранить информацию). Данный сервер сопоставляет имя сайта из запроса с IP адресом, так же перед этим компьютер проверяет наш кэш браузера, вдруг мы уже были на данном сайте. Если в DNS сервере нет информации о данном запросе, то он отправляет запрос на root DNS сервер. Это сервер уровнем выше- над всеми серверами, он указывает к какому корневому DNS серверу обратиться (TLD- Top level Domain server) – .com, .net, .ru и т.д. TLD делаться на gTLD и cсTLD. Одни не привязаны к региону или стране, другие привязаны. TLD серверы уже скажут IP адрес или не скажут, если сайта нет. Далее получив адрес, можно переходить на сайт и загружать ресурс. Так же resolver DNS записывает у себя данные названия сайта и адреса для ускорения процесса в будущем.

**HTTP|HTTPS:**

HTTP – это Hyper text protocol( протокол передачи гипертекста). Гипертекст появился довольно давно в литературе, фактически это текст с ссылками на другие страницы или события в книге. Когда мы запускаем сайт или что то кликаем на сайте и дальше перебросит нас – это гипер ссылка. Когда нам загружается сайт, мы получаем страницу, написанную на языке HTML( Hypertext markup Language – язык гипертекстовой разметки). Так же надо отметить, что http это клиент-сервер протокол, иными словами есть наш компьютер (клиент), есть наш запрос на информацию, есть потенциальный сервер, содержащий информацию. Мы делаем запрос по протоколу и получаем информацию с сервера, которая содержится на нашему экране. После http мы обычно используем URL (Uniform Resource Locator) – унифицированный указатель ресурса. Каждый наш запрос через **протокол http** содержит в себе основные стандартные строки:

**Стартовая строка** с содержанием URL адреса, и методом его запроса(GET, Post,Put, Delete и т.д.). Сервер нам отвечает, что все ок и присылает код 200, если все плохо, то присылает 404:( и т.д.

**Заголовок** содержит в себе информацию технического характера о том, какой у нас браузер, какая у нас система, какой у нас язык, пароли, логины и т.д.

**Тело сообщения:** здесь мы передаем серверу информацию, что мы от него хотим и он в ответ отправляет на наш запрос нужную информацию. Все действия передаются от клиента к серверу через текстовый файл т.к. данный протокол был придуман весьма давно – в далеком 1990 году и запушен в 1992. Фактически тогда еще не было интернета в том виде, как мы привыкли его видеть. Это была просто большая компьютерная сеть.

Поэтому можно было перехватить или параллельно обработать такой тестовый файл. В 90-х – 2000х это было не критично, люди мало передавали информации, требующей секретности. Далее появился http**s –** означающий http secure. Данный протокол шифрует данные через механизм SSL(secure sockets level)/TLS(transport layer security). Каждый сервер, работающий через данный протокол, имеет ssl сертификат, который выдает специальной сертификационное бюро. Данный сертификат подтверждает нам, что сайт надежный. На данный момент, используется TLS 1.2-1.3 протоколы.

Разница между версиями http: в протоколе версии 1.0 запросы передаются через текстовый формат на сервер. После того, как запрос обработан, сервер передал данные пользователю, соединение закрывается. Поэтому требовалось много ресурсов для обновления сайта при просмотре и т.д., скорость низкая была то время. Поэтому придумали версию 1.1 – в ней уже есть постоянное соединение между клиентом и сервером. Но было узкое место – запросы шли друг за другом и если не мог один выполниться, то остальные ожидали, а открыть новое ТСP соединение нельзя – нужны ресурсы и есть ограничения.

Поэтому придумали версию 2.0 – в ней уже запросы и ответы передаются в двоичном коде, что ускоряет процесс обработки. Данные разбиваются на мелкие пакеты. Данные передаются несколькими потоками. Так же появилось шифрование – что весьма актуально в современном интернете.