мрежови видове протоколи http

Мрежов протокол е система от стандартни правила за обмяната на информация между процеси или компютри, които са свързани в една компютърна мрежа, и то по начин, гарантиращ успешната връзка между два или повече комуникационни апарата (крайни устройства). Комуникацията в тази мрежа се базира на множество протоколи с различни функции и се осъществява посредством обмяната на съобщения (пакети). Описаните в протокола правила дефинират каква информация се предоставя в пакетите и в какъв формат, за да се приеме от комуникационните партньори.

Най-често използваните протоколи са протоколите за пренасяне на данни през мрежа. Съществуват 2 модела за класификация на тези протоколи – OSI и TCP/IP.

Типично съдържание на пакет

Пакетите съдържат:

Адресите (IP адрес или хардуерен) на подателя и получателя

тип на пакета (например дали се отнася за създаване или прекратяване на връзка)

дължина на пакета

сума за проверка на целостта на пакета (на английски: checksum)

Тази информация се поставя преди или след данните в даден пакет. Освен това някои протоколи дефинират определени секвенции от съобщения, които трябва да бъдат изпълнени, за да бъде създадена връзката между комуникационните партньори. Тази информация, наричана оувърхед (на английски: Overhead), натоварва допълнително мрежата, но е необходима за правилното доставяне на пакетите. Съществуват олекотени протоколи с по-малко допълнителна информация, но те са считани за несигурни и се ползват само при определени обстоятелства.

Дефиниции в мрежовите протоколи

Брой на комуникационните партньори: Ако получателят е само един – комуникацията е от тип уникаст (на английски: Unicast), при комуникация между повече партньори – тип мултикаст (на английски: Multicast).

Ако информацията се предава само в едната посока, връзката се нарича симплекс (на английски: Simplex). Ако съобщенията в двете посоки се редуват последователно, връзката е от тип полудуплекс (англ. Halfduplex). Ако информацията се предава едновременно и в двете посоки на комуникация, връзката се нарича пълен дуплекс (на английски: Fullduplex).

Роля на партньорите: ако са с еднакви роли, връзката е симетрична (също peer-to-peer), в противен случай – асиметрична. Типичен пример за асиметрична връзка е модела клиент-сървър (на английски: Client-Server-System), при който един от партньорите приема и обработва запитвания от много клиенти, които са инициаторите на връзката.

Ако при дадено запитване се изисква отговор (например за създаване на връзка), комуникацията се нарича синхронна, в противен случай – асинхронна.

Информацията може да е в пакети или във вид на непрекъснат поток от отделни знаци (на английски: streaming).

При протоколите с повече оуверхед се ползват определени поредици от пакети за дефиниране на начало и край на връзка. Такива протоколи осигуряват транспорта на пакети в определен ред и повторното им пращане, ако не стигнат до получателя. Други протоколи не предлагат такива функции, но имат по-малко оувърхед и комуникацията е по-бърза.

Функции на модерните протоколи

Сигурно и надеждно осъществяване на връзка между комуникационните партньори.

Надеждно доставяне на пакетите на желаните получатели.

Повторно пращане на неполучени пакети.

Ползване на контролната сума за проверка на целостта на пакета.

Събирането на пакетите в правилната поредица за изграждането на цялостната информация (като се има предвид, че информацията се разделя при транспортиране на множество малки пакети).

Възпрепятстване на нежелан достъп и промяна на информацията (кодиране).

http заявки

HTTP заявките имат 3 основни елемента: метод, Request-URI и header-полета.

Методът описва вида на HTTP заявката, изпратена от клиента. Най-често използваните методи са GET и POST. Чрез GET клиентът изисква някакъв ресурс от Web сървъра. POST служи за предаване на данни към сървъра и извличане на ресурс. Имената на методите в HTTP заявките се изписват винаги с главни букви.

Идентификаторът Request-URI определя ресурса, над който ще оперира заявката. В частта Request-URI могат да се използват два вида идентификатори:

- URI идентификатор (Uniform Resource Identifier)

- релативен път спрямо главната директория на Web-сървъра

Един URI идентификатор може да бъде или URL адрес (Uniform Resource Location, например http://www.nakov.com/inetjava/index.html), т.е. да е идентификатор на ресурс, зададен чрез уникалното си местоположение или URN име (Uniform Resource Name, например urn:isbn:954-8905-06-X), т.е. да е идентификатор на ресурс, зададен чрез уникалното си име по даден URN namespace идентификатор (за нашия пример това е идентификатора isbn). В практиката URN схемата за идентификация на ресурс почти не се използва при HTTP заявки.

Релативният път спрямо главната директория на Web-сървъра задава местоположението на ресурс в рамките на текущия Web-сървър. Това е частта от URL, която стои след името на хост-а (сървъра) в URL идентификатора. Например един такъв релативен път може да бъде идентификаторът /inetjava/index.html.

Фрагментът HTTP/1.1 с който завършва първият ред на HTTP заявката задава версията на HTTP протокола, която ще бъде използвана.

Header-полетата от заглавната част задават допълнителни параметри на заявката и определят различни изисквания относно ресурса, който се очаква да бъде върнат от Web-сървъра.

Празният ред определя края на заявката.

Kлиент сървър комуникация

Клиент – сървър е тип мрежова архитектура, която отделя клиента от сървъра и най-често се използва в компютърни мрежи.[1] Всеки клиент или сървър, свързан с мрежата, може също така да бъде използван като възел.

Най-елементарната типова клиент – сървър архитектура се състои от две части – от едната страна е сървърът, от другата страна е клиентът. Постигането на определен резултат при използване винаги е свързано с комуникация между двете страни

Реализация на чат приложение в модел клиент сървър kомуникация

Модел клиент-сървър е разпределено приложение структура, която разделя задачи или натоварвания между доставчиците на ресурс или услуга, наречена сървъри, и заявители на услуги, извикани клиенти.[1] Често клиентите и сървърите комуникират през a компютърна мрежа на отделен хардуер, но и клиентът, и сървърът могат да се намират в една и съща система. Сървър домакин изпълнява една или повече сървърни програми, които споделят своите ресурси с клиенти. Клиентът не споделя никой от своите ресурси, но иска съдържание или услуга от сървър. Следователно клиентите инициират комуникационни сесии със сървъри, които очакват входящи заявки. Примери за компютърни приложения, които използват модела клиент-сървър, са електронна поща, мрежов печат и Световната мрежа.