ANALYSIS OF THE OPERATING PRINCIPLES AND EFFICIENCY OF THE TCP AND UDP TRANSPORT PROTOCOLS*

DESISLAV M. HRISTOV

ABSTRACT: *This paper examines the main characteristics, operating principles, and applications of the transport protocols TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) within the TCP/IP architecture. Their key features are presented in terms of connection establishment, data transmission reliability, flow control, error handling, and communication speed. A comparative analysis highlights the advantages, limitations, and application areas of both protocols. Examples of their use in various network services and applications are discussed, where either reliable information delivery or minimal transmission latency is a primary requirement. The aim of the paper is to clarify the functional differences between TCP and UDP and to demonstrate the importance of selecting the appropriate transport protocol according to the specific requirements of a network application.*

KEYWORDS: *TCP, UDP, TCP/IP, transport layer, network protocols, reliability, data transmission*

DOI: https://doi.org/10.46687/MT26DMH

АНАЛИЗ НА ПРИНЦИПИТЕ НА РАБОТА И ЕФЕКТИВНОСТТА НА ТРАНСПОРТНИТЕ ПРОТОКОЛИ TCP И UDP

ДЕСИСЛАВ М. ХРИСТОВ

АБСТРАКТ: *Настоящият доклад разглежда основните характеристики, принципите на функциониране и приложението на транспортните протоколи TCP (Transmission Control Protocol) и UDP (User Datagram Protocol) в архитектурата TCP/IP. Представени са техните основни особености по отношение на установяването на връзка, надеждността на предаването на данни, контрола на потока, управлението на грешките и скоростта на комуникация. Направен е сравнителен анализ на двата протокола, като са откроени техните предимства, ограничения и области на приложение. Разгледани са примери за използването им в различни мрежови услуги и приложения, при които водещо значение имат или надеждното доставяне на информацията, или минималното закъснение при предаването ѝ. Целта на доклада е да изясни функционалните различия между TCP и UDP и да покаже значението на правилния избор на транспортен протокол според конкретните изисквания на мрежовото приложение.*

1 Въведение

Съвременните компютърни мрежи представляват основа за обмен на информация между различни устройства, системи и приложения. За осъществяването на този обмен се използват мрежови протоколи, които определят правилата и механизмите за предаване и получаване на данни. В архитектурата TCP/IP особено важна роля има транспортният слой, който осигурява комуникацията между приложенията, работещи на крайни устройства. Сред основните протоколи на транспортния слой са TCP (Transmission Control Protocol) и UDP (User Datagram Protocol). Въпреки че изпълняват сходна основна функция, двата протокола се различават съществено по начина, по който организират предаването на

* Докладът е частично финансиран по проект № РД-08-113/05.02.2026 г. „Научна конференция с международно участие МАТТЕХ 2026 – ФТН“

информация. TCP осигурява надеждно предаване на данни чрез механизми за установяване на връзка, потвърждение на получаването и повторно изпращане на загубена информация. UDP, от своя страна, предлага по-опростен механизъм за предаване без предварително установяване на връзка, което го прави подходящ за приложения, при които бързината и ниското закъснение са от съществено значение. Изборът между TCP и UDP зависи от специфичните изисквания на съответното приложение, включително необходимата надеждност, допустимото закъснение и ефективността на комуникацията. Поради това познаването на особеностите на двата протокола е важно за разбирането на принципите на функциониране на компютърните мрежи и на различните услуги, предоставяни чрез интернет. На фиг.1 е показано позиционирането на протоколите TCP и UDP



Фиг. 1. Позициониране на протоколите TCP и UDP в архитектурата на TCP/IP

2 Ролята на TCP и UDP в архитектурния модел TCP/IP

Съвременните интернет комуникации се осъществяват чрез съвкупност от протоколи, които определят начина, по който устройствата обменят информация. За да бъде комуникацията организирана и предвидима, различните функции се разделят между отделни слоеве. Всеки слой изпълнява определени задачи и използва услугите на слоя под него. Този подход улеснява разработването на мрежови приложения, съвместимостта между различни системи и откриването на проблеми при предаването на данни. В архитектурата TCP/IP транспортният слой осигурява комуникация между приложения, работещи на различни крайни устройства. Докато интернет слоят отговаря за логическото адресиране и пренасянето на IP пакетите между мрежите, транспортният слой позволява данните да бъдат насочени към конкретното приложение на крайното устройство. За тази цел се използват транспортни портове. Основните транспортни протоколи, разглеждани в настоящия доклад, са Transmission Control Protocol (TCP) и User Datagram Protocol (UDP). Те изпълняват сходна обща задача – пренасяне на данни между приложения – но използват различни механизми. TCP предоставя надежден, ориентиран към връзка пренос, докато UDP изпраща дейтаграми без предварително установяване на транспортна връзка и без гаранция за доставката им. Изборът между двата протокола зависи от изискванията на конкретното приложение [2, 3].

Моделът TCP/IP описва архитектурата на протоколите, използвани в интернет. Обикновено той се представя чрез четири слоя: приложен, транспортен, интернет и слой за мрежов достъп. Всеки от тях има определена роля в процеса на комуникация. Приложният слой обединява протоколите и механизмите, чрез които мрежовите приложения предоставят услуги на потребителите или обменят информация помежду си. Примери за

такива протоколи са HTTP и HTTPS за уеб комуникация, SMTP за изпращане на електронна поща и DNS за преобразуване на домейн имена в IP адреси. Транспортният слой осигурява обмена на данни между приложенията на крайните системи. TCP и UDP работят на това ниво. Чрез номерата на портовете се определя към кое приложение или мрежова услуга трябва да бъдат предадени получените данни. Портът не заменя IP адреса: IP адресът идентифицира мрежовия интерфейс или крайното устройство в рамките на IP комуникацията, а портът подпомага разграничаването на комуникационните крайни точки на транспортно ниво. Интернет слоят използва IP за логическо адресиране и маршрутизация на пакетите през свързани мрежи. IP предоставя механизъм за доставка на пакети до определен адрес, но сам по себе си не гарантира, че пакетът ще достигне до получателя, нито че пакетите ще пристигнат в първоначалния ред. Допълнителните механизми, необходими на приложенията, се осигуряват от транспортния протокол или от други протоколи и механизми в системата. Слоят за мрежов достъп обхваща механизмите за предаване на данни през конкретната мрежова технология. Той включва взаимодействието с локалната мрежова среда и физическото предаване на сигналите. Примери за технологии, използвани на това ниво, са Ethernet и Wi-Fi. При изпращане на данни приложението предава информацията към транспортния слой. TCP или UDP добавя необходимата транспортна информация, след което данните се предават към IP за адресиране и маршрутизация. На приемащото устройство процесът се извършва в обратна посока: IP обработва получените пакети, транспортният слой използва информацията за протокола и портовете, а данните се предават към съответното приложение.

Мрежовите портове са логически номера, които се използват от транспортните протоколи TCP и UDP за разграничаване на различни комуникационни крайни точки. Номерът на порт сам по себе си не представлява физически конектор и не означава непременно, че съответната услуга е достъпна. Той е част от информацията, чрез която операционната система обработва входящия и изходящия трафик. TCP и UDP използват отделни пространства от номера на портове. Поради това един и същ номер може да се използва независимо от двата протокола, когато това е предвидено от съответните приложения и конфигурации. В практиката често се срещат стандартно асоциирани портове: например TCP порт 22 се свързва със SSH, TCP порт 25 – със SMTP, а DNS обичайно използва порт 53 както с UDP, така и с TCP според вида на заявката и условията на комуникацията. При разглеждането на мрежовите услуги е важно да се прави разлика между номер на порт и самия протокол или услуга. Портът обозначава транспортна крайна точка, TCP и UDP определят начина на транспортен обмен, а приложният протокол задава правилата за интерпретиране на съобщенията. Например при SSH TCP осигурява транспортната връзка, докато SSH определя механизмите за защитен отдалечен достъп. Съчетанието от IP адрес, транспортен протокол и номер на порт позволява да се разграничават комуникационните крайни точки. Това е важно за работата на сървъри, които могат едновременно да обслужват множество клиенти и различни услуги.

Макар TCP и UDP да работят на един и същ слой от архитектурата TCP/IP, техните характеристики са различни и определят подходящите области на приложение. TCP включва механизми за потвърждение, повторно предаване на загубени данни, подреждане на байтовия поток и управление на потока и претоварването. Тези механизми са полезни, когато приложението изисква надеждно и подредено предаване на информация. UDP има по-малко функции на транспортно ниво. Той позволява изпращането на отделни дейтаграми без предварително установяване на връзка и не осигурява вградено повторно предаване или гарантирано подреждане. Това може да бъде полезно за приложения, които

предпочитат ограничено протоколно натоварване и могат да обработват загуба на данни или да реализират собствени механизми за надеждност. Не е правилно обаче UDP автоматично да се определя като по-бърз във всички ситуации. Реалната производителност и закъснение зависят от мрежовите условия, размера и честотата на съобщенията, поведението на приложението и използваните допълнителни механизми. По същия начин TCP не е подходящ за всяка задача само защото предоставя надеждност. Изборът трябва да се основава на изискванията към доставката, последователността, времето за реакция и обработката на загубени данни. Разглеждането на архитектурата TCP/IP, ролята на портовете и функциите на транспортния слой създава необходимата основа за по-подробното представяне на TCP и UDP в следващите глави. На фиг. 2 е показана ролята на транспортния слой [2, 3].



Фиг. 2. Роля на транспортния слой при осъществяване на комуникация

3 Принцип на работа и характеристики на протокол TCP

Transmission Control Protocol (TCP) е транспортен протокол от архитектурата TCP/IP, предназначен да осигурява надежден и подреден обмен на данни между приложения, работещи на различни крайни устройства. За разлика от UDP, TCP е ориентиран към връзка: преди предаването на данни страните установяват логическа транспортна връзка, а по време на обмена протоколът използва механизми за потвърждение, контрол на последователността и повторно предаване при определени случаи на загуба. TCP предоставя на приложението поток от байтове. Това означава, че протоколът гарантира подреденото предаване на данните в рамките на връзката, но не запазва непременно границите на отделните съобщения, които приложението подава към транспортния слой. Ако приложението изпрати информация на няколко отделни части, получаващата страна трябва да използва правилата на съответния приложен протокол, за да определи къде завършва едно съобщение и започва следващото. Една от основните характеристики на TCP е надеждността. Протоколът използва номерация на предаваните данни и потвърждения от приемащата страна, за да проследява кои части от потока са получени. Ако данни се загубят по пътя, TCP може да ги предаде повторно. Освен това протоколът поддържа механизми за контрол на потока и контрол на претоварването. Те помагат да се ограничи изпращането на данни според възможностите на получателя и състоянието на мрежата. Тези механизми правят TCP подходящ за приложения, при които е важно информацията да бъде доставена правилно и в определен ред. Примери са зареждането на уеб страници чрез HTTP/1.1 и HTTP/2, прехвърлянето на файлове и отдалеченото администриране чрез SSH. При тези приложения загубата или разместването на данни без коригиращи механизми може да доведе до неправилно обработване на информацията. Преди да започне обичайният обмен на данни, TCP страните установяват връзка чрез процедура, известна като Three-Way

Handshake, или тристранно ръкостискане. Чрез нея клиентът и сървърът обменят контролни сегменти, синхронизират началните номера на последователност и потвърждават, че комуникацията може да бъде започната. Процедурата включва три основни стъпки:

SYN – клиентът изпраща сегмент със зададен SYN флаг. В него се посочва началният номер на последователност, който клиентът ще използва за изпращаните данни.

SYN-ACK – ако сървърът приема заявката, той отговаря със сегмент, който съдържа SYN и ACK флагове. Сървърът посочва собствен начален номер на последователност и потвърждава получения SYN от клиента.

ACK – клиентът изпраща потвърждение на началния номер на последователност на сървъра. След успешното приключване на обмена връзката е установена и може да започне предаването на данни.

Тази процедура позволява на двете страни да синхронизират необходимата информация за проследяване на потока. Тя не е самото предаване на приложните данни, а предварителен етап за установяване на транспортната връзка. След него TCP продължава да използва номера на последователност и потвърждения, за да управлява обмена. Тристранното ръкостискане има значение и за обработването на закъснели или дублирани сегменти от предишни опити за комуникация. Чрез обмена на актуални параметри страните потвърждават, че могат да участват в новата връзка. Процедурата обаче не осигурява криптиране или удостоверяване на самоличността сама по себе си. Ако приложението изисква защита на съдържанието и проверка на идентичността на отсрещната страна, са необходими допълнителни механизми, например TLS или защитен приложен протокол. При предаване през IP мрежа отделни пакети могат да бъдат загубени, дублирани или доставени в различен ред от този, в който са изпратени. IP не предоставя на транспортния слой гаранция за надеждна доставка. TCP компенсира част от тези ограничения, като номерира байтовете в потока и използва потвърждения от получателя. Потвърждението показва до коя част от потока получателят е приел данните последователно. Ако изпращащата страна не получи очаквано потвърждение в рамките на определен период или установи загуба по друг поддържан механизъм, тя може да изпрати отново липсващите данни. Приемащата страна използва номерата на последователност, за да възстанови правилния ред на потока и да не предава повторно получени данни като ново съдържание на приложението. TCP осигурява поток от байтове, а не гарантира, че всяко изпращане от приложението ще съответства на един отделен сегмент или едно отделно получаване. Операционната система може да разделя или обединява данните при предаването им. Поради тази причина приложните протоколи трябва самостоятелно да определят структурата и границите на своите съобщения. Надеждността на TCP не означава, че предаването винаги е незабавно. При загуба на сегменти повторното им изпращане може да забави получаването на следващите данни от потока. Това е важен компромис: TCP се стреми да предостави пълен и подреден поток, дори когато мрежата не доставя всички сегменти без проблеми. TCP използва два различни, но взаимно допълващи се механизма: контрол на потока и контрол на претоварването. Те имат различни цели и не трябва да се разглеждат като едно и също действие. Контролът на потока предпазва получателя от прекомерно количество входящи данни. Получаващата страна съобщава какъв обем данни може да приеме в дадения момент, като по този начин изпращачът може да съобрази количеството данни, което изпраща без допълнително потвърждение. Това е важно, когато устройствата имат различни изчислителни възможности или когато приложението на получателя обработва данните по-бавно от тяхното постъпване. Контролът на претоварването има за цел да ограничава натоварването върху самата мрежа. Ако се

изпраща повече трафик, отколкото междинните мрежови устройства и връзки могат да обработят, може да се увеличат закъсненията и загубите на пакети. TCP използва алгоритми за контрол на претоварването, чрез които адаптира скоростта на изпращане спрямо наблюдаваното състояние на мрежата. Разликата е съществена: контролът на потока се съобразява с капацитета на приемащата страна, докато контролът на претоварването се стреми да предотврати прекомерно натоварване на мрежовия път. Заедно тези механизми подпомагат стабилната работа на TCP при различни мрежови условия. На фиг. 3 са показани основните етапи на функциониране на TCP.[2]

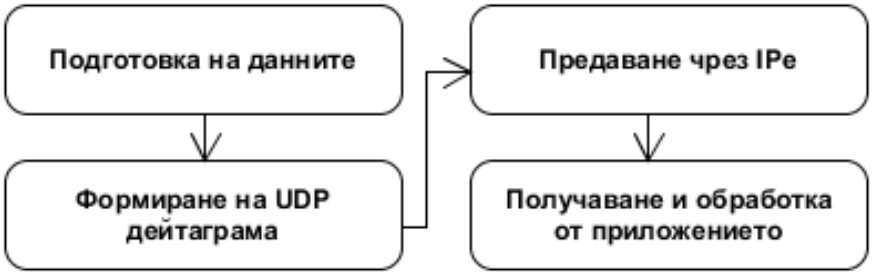


Фиг. 3. Основни етапи на функциониране на TCP

4 Принцип на работа и характеристики протокол UDP

User Datagram Protocol (UDP) е транспортен протокол от архитектурата TCP/IP, който позволява обмен на данни между приложения без предварително установяване на транспортна връзка. За разлика от TCP, UDP не поддържа механизми за потвърждение на получаването, повторно предаване на изгубени дейтаграми или гарантирано подреждане на данните. Изпращачът оформя съобщението в UDP дейтаграма и я предава към IP слоя за пренасяне през мрежата. UDP има сравнително проста структура и малко протоколно натоварване. Заглавната част на UDP е с фиксирана дължина от 8 байта и съдържа четири полета: порт на източника, порт на получателя, дължина и контролна сума. Портовете позволяват на приемащата система да предаде дейтаграмата към съответното приложение. Полето за дължина указва размера на UDP заглавната част и данните, а контролната сума служи за откриване на определени грешки при предаването. Конкретното използване и задължителност на контролната сума зависят от версията на IP и съответните правила на протокола. Тъй като UDP не установява връзка, преди изпращането не е необходим обмен на контролни сегменти, подобен на TCP Three-Way Handshake. Това намалява броя на съобщенията, нужни за започване на комуникацията. Протоколът обаче не може самостоятелно да гарантира, че дейтаграмата ще достигне до получателя, че ще пристигне само веднъж или че ще бъде обработена в реда на изпращане. Ако приложението се нуждае от подобни свойства, то трябва да ги реализира на приложно ниво или да използва друг протокол, който ги предоставя. При UDP данните се изпращат като отделни дейтаграми. Всяка дейтаграма е самостоятелна единица, за разлика от TCP, който представя данните към приложението като поток от байтове. Границите на UDP съобщенията се запазват на транспортно ниво, но това не означава, че мрежата гарантира тяхната доставка. UDP разчита на IP за пренасянето на дейтаграмите между мрежите. По време на комуникацията

дейтаграми могат да бъдат загубени, дублирани или получени в различен ред. UDP не изпраща автоматично повторно изгубените съобщения и не подрежда получените дейтаграми. Освен това протоколът не включва собствен механизъм за контрол на потока или контрол на претоварването. Това не означава, че приложенията върху UDP непременно работят без такива механизми: ако се изискват, те трябва да бъдат реализирани от приложението или от протокол от по-високо ниво. Липсата на вградени механизми за надеждност може да бъде ограничение за приложения, при които всяка част от информацията трябва да бъде получена правилно. В същото време тя може да бъде предимство, когато по-важно е актуалните данни да пристигат навреме, отколкото изгубените данни да бъдат повторно изпращани със закъснение. Например при гласова комуникация кратка загуба на аудио данни може да бъде по-малко смущаваща от продължително забавяне, причинено от повторно предаване. Реалното качество обаче зависи от приложението, мрежовите условия и използваните механизми за обработка на загубите. Основните предимства на UDP са простата структура, малката заглавна част и възможността за изпращане на данни без предварително установяване на връзка. Това го прави подходящ за кратки заявки и отговори, както и за приложения, при които се изпращат чести актуализации и закъснението е важен фактор. Друго свойство е, че UDP позволява изпращането на съобщения до множество получатели чрез механизми на IP като broadcast или multicast, когато мрежата и приложението ги поддържат. Това може да бъде полезно при определени услуги за откриване на устройства, разпространение на информация или групова комуникация. Използването на такива механизми зависи от мрежовата среда и не е гарантирано във всяка интернет връзка. Най-същественото ограничение е липсата на вградени гаранции за доставка, подредба и защита от претоварване. Приложението трябва да прецени дали може да работи при загуба на дейтаграми или дали са необходими допълнителни механизми. Затова UDP не трябва да се определя като универсално по-бърз или по-добър от TCP. Той предоставя по-опростен транспортен механизъм, а пригодността му зависи от задачата. На фиг. 4 са показани основните етапи на функциониране на UDP.[3]



Фиг. 4. Основни етапи на функциониране на UDP

UDP се използва в различни интернет услуги и приложения. Един от примерите е DNS (Domain Name System), който преобразува домейн имена в IP адреси. Обичайните DNS заявки често използват UDP порт 53, тъй като обменът обикновено се състои от кратка заявка и отговор. DNS може да използва и TCP, например при някои по-големи отговори, трансфер на зони и други случаи, определени от протокола. Следователно DNS не е услуга, която използва единствено UDP. Друг пример са приложенията за гласова и видео комуникация в реално време. При такива приложения навременното получаване на текущите аудио- или видео данни често е по-важно от повторното предаване на вече остаряла информация. UDP може да намали транспортните забавяния, но качеството на

комуникацията зависи и от кодека, буферирането, мрежовото закъснение, загубата на пакети и механизмите за адаптация. UDP се използва и в онлайн игрите, при които често се изпращат актуализации за позицията и състоянието на игровите обекти. Ако някоя актуализация бъде изгубена, следваща може бързо да предостави по-нова информация. Това не означава, че всички онлайн игри използват само UDP: изборът зависи от архитектурата на конкретната игра и от изискванията към различните видове данни. Съвременен пример за използване на UDP като основа за по-сложен транспорт е QUIC, който се използва от HTTP/3. QUIC работи върху UDP, но реализира собствени механизми за надеждна доставка, криптографска защита и контрол на претоварването. Това показва, че UDP може да служи като базов транспорт за протоколи, които добавят функции, липсващи в самия UDP.

5 Сравнителен анализ на TCP и UDP

TCP и UDP изпълняват функцията на транспортни протоколи в архитектурата TCP/IP, но предлагат различни възможности за обмен на данни. TCP е ориентиран към връзка и осигурява надежден, подреден поток от байтове. UDP изпраща отделни дейтаграми без предварително установяване на връзка и не предоставя вградени гаранции за доставка и последователност. Поради това изборът между двата протокола трябва да се прави според изискванията на приложението, а не само според броя на функциите, които протоколът поддържа. Основните критерии за сравнение са начинът на установяване на комуникацията, надеждността, запазването на реда на данните, протоколното натоварване, управлението на потока и претоварването, както и типичните приложения. Тези критерии показват как характеристиките на транспортния слой влияят върху поведението на услугите, които използваме в интернет. Таблица 1 обобщава основните разлики между TCP и UDP

Таблица 1. Основни разлики между TCP и UDP

Критерий	TCP	UDP
Установяване на връзка	Установява логическа връзка преди обмена на данни	Не изисква предварително установяване на връзка
Тип на предаването	Поток от байтове	Отделни дейтаграми
Надеждност	Използва потвърждения и повторно предаване при загуба	Не осигурява вградено потвърждение или повторно предаване
Подредба на данните	Предоставя подреден поток към приложението	Не гарантира реда на пристигане на дейтаграмите
Контрол на потока	Поддържа механизъм за контрол на потока	Няма вграден механизъм за контрол на потока
Контрол на претоварването	Поддържа механизми за контрол на претоварването	Самият UDP не предоставя такъв механизъм; приложението или протокол от по-високо ниво може да го реализира
Размер на заглавната част	Минимално 20 байта без опции	8 байта
Протоколно натоварване	По-голямо заради състоянието на връзката и контролните механизми	По-малко, поради по-простата структура
Типични приложения	Уеб комуникации чрез HTTP/1.1 и HTTP/2, SSH, прехвърляне на файлове, електронна поща	DNS заявки, комуникации в реално време, онлайн игри и протоколи, изградени върху UDP

Посочените размери се отнасят до транспортните заглавни части: TCP има минимална заглавна част от 20 байта, която може да бъде по-голяма при наличие на опции, докато UDP заглавната част е 8 байта. Това е една от причините UDP да има по-малко транспортно натоварване, но само по този показател не може да се определи цялостната производителност на дадено приложение. Една от най-съществените разлики между TCP и UDP е начинът, по който се обработва загубата на информация. TCP се стреми да предостави на приложението пълен и подреден поток от данни. При загуба на сегмент протоколът може да го предаде повторно, а получателят възстановява правилната последователност на потока. Това е особено важно при прехвърляне на файлове, зареждане на уеб съдържание и отдалечено управление, където липсващите или разместени данни могат да доведат до неправилно обработване. При UDP всяка дейтаграма се обработва самостоятелно и протоколът не предприема автоматично повторно предаване. Това може да е недостатък, когато всяко съобщение трябва да бъде получено, но може да бъде приемливо при приложения, които обменят чести актуализации. Например при гласова комуникация повторното предаване на вече остарял аудио пакет може да е по-малко полезно от своевременното възпроизвеждане на следващите данни. Тук е важно да се направи разграничение между надеждност и навременност. TCP предоставя надеждност на транспортно ниво, но при загуби повторното предаване може да увеличи закъснението. UDP не гарантира по-ниско закъснение във всички мрежи, но избягва задължителните механизми за установяване на връзка и повторно предаване, присъщи на TCP. Реалното поведение зависи от мрежовите условия и от начина, по който приложението обработва получените данни. TCP поддържа състояние на връзката и използва контролни съобщения, потвърждения и механизми за управление на предаването. Това създава допълнителен протоколен обмен, но осигурява надеждност и адаптиране към възможностите на получателя и състоянието на мрежата. Контролът на потока предотвратява изпращането на повече данни, отколкото получателят може да обработи, а контролът на претоварването ограничава риска от прекомерно натоварване на мрежовия път. UDP има по-малка заглавна част и не изисква установяване на връзка. Това може да намали разходите за обмен при кратки съобщения и приложения, които изпращат чести дейтаграми. Въпреки това по-малкото функции на самия UDP не означават, че цялото приложение непременно ще използва по-малко ресурси или ще работи по-бързо. Ако върху UDP се изграждат собствени потвърждения, повторни предавания, криптиране и контрол на претоварването, тези функции също изискват ресурси и обмен на данни. Затова при оценката на производителността трябва да се разглежда цялата комуникационна система: приложният протокол, размерът на съобщенията, честотата на изпращане, загубите, закъснението и наличната пропускателна способност. Инструментите за анализ на трафика на локална мрежа са разгледани в [1]. TCP и UDP са различни инструменти за различни изисквания, а не универсална класация от „по – бавен“ и „по – бърз“ протокол. Сравнението показва, че TCP и UDP са предназначени за различни комуникационни потребности. TCP е подходящ, когато е необходим надежден и подреден поток от данни, а UDP предоставя по-прост обмен на отделни дейтаграми с по-малко вградени транспортни механизми. Не може да се твърди, че единият протокол е по-добър във всички случаи. При приложения, при които целостта и последователността на данните са водещи, TCP предоставя полезни гаранции. При приложения, чувствителни към закъснение или използващи собствени механизми за управление на обмена, UDP може да бъде по-подходящ. В крайна сметка решението се определя от изискванията към надеждността, времето за реакция, мрежовото натоварване и архитектурата на приложението. Този сравнителен анализ създава основа за следващата

глава, в която ще бъдат разгледани конкретни примери за използване на TCP и UDP в интернет услугите [2, 3].

6 Практическо приложение на TCP и UDP в интернет комуникациите

TCP се използва широко от приложения, при които е важно данните да бъдат предадени надеждно и в правилния ред. Типичен пример са уеб комуникациите чрез HTTP/1.1 и HTTP/2. При тези версии на HTTP транспортният обмен обичайно се осъществява чрез TCP. Протоколът осигурява подреден поток от байтове, върху който HTTP реализира правилата за заявките, отговорите и съдържанието на уеб ресурсите. Когато потребител отвори уеб страница, браузърът обменя съобщения със сървъра. На транспортно ниво TCP управлява връзката, проследява получаването на данните и при необходимост предава повторно изгубени части от потока. Това позволява на приложението да обработва информацията в правилната последователност. При HTTPS към комуникацията се добавя TLS, който осигурява криптографска защита на данните и проверка на сертификата на сървъра според използваната конфигурация. TCP сам по себе си не криптира съдържанието. Друг пример е SSH (Secure Shell), използван за защитен отдалечен достъп и администриране на системи. SSH обичайно използва TCP порт 22. TCP предоставя надежден транспортен канал, а SSH реализира функции като удостоверяване, криптиране и изпълнение на отдалечени команди. Това разграничение показва, че транспортният протокол и приложният протокол имат различни, но взаимно допълващи се роли. TCP се използва и при прехвърляне на файлове и електронна поща. При тези задачи е важно получените данни да съответстват на изпратените. Например SMTP използва TCP за обмен на съобщения между пощенски системи, а различни протоколи за достъп до пощенски кутии също могат да използват TCP. Надеждният транспорт намалява необходимостта приложението да реализира самостоятелно механизми за възстановяване на загубени части от потока. Един от широко разпространените примери за UDP е системата DNS (Domain Name System). DNS преобразува домейн имена, например имена на уеб сайтове, в IP адреси, необходими за установяване на мрежова комуникация. Обичайните DNS заявки често се изпращат чрез UDP порт 53. Тъй като много от тях се състоят от кратка заявка и отговор, UDP позволява обменът да започне без предварително установяване на TCP връзка. DNS обаче не използва само UDP. TCP също е част от DNS спецификациите и се използва в определени случаи, например при трансфер на DNS зони, както и когато условията на протокола налагат това. Затова при описание на DNS е по-точно да се посочи, че UDP е често използван за обичайни заявки, но TCP също има важна роля. UDP се използва и при гласови и видео приложения в реално време. При разговор през интернет приложението изпраща последователност от малки пакети с аудио- или видео данни. Ако пакет пристигне твърде късно, той може да има малка практическа стойност, тъй като възпроизвеждането вече е продължило. В подобни ситуации приложението може да предпочете навременната доставка на следващите данни пред повторното изпращане на вече закъснели данни. Използването на UDP не гарантира само по себе си високо качество на разговор или малко закъснение. Качеството зависи и от пропускателната способност, загубата на пакети, мрежовото закъснение, използвания кодек, буферирането и механизмите за адаптация. Приложенията често включват допълнителни средства за оценка на състоянието на мрежата и за компенсиране на загуби и колебания в закъснението. Онлайн игрите са пример за приложения, при които навременната информация често е от особено значение. Клиентите и сървърът могат да обменят данни за позицията на играчите,

състоянието на обектите и действията в игровия свят. При чести актуализации някои по-стари съобщения могат да загубят значението си, ако междувременно са получени по-нови данни. Затова UDP може да бъде подходящ за определени видове игрови трафик. При UDP обаче разработчиците трябва да отчетат възможността за загуба, дублиране или промяна на реда на дейтаграмите. Ако конкретна информация не бива да се губи, играта може да реализира собствени потвърждения или да използва отделен канал или протокол за нея. Следователно изборът на UDP не премахва нуждата от надеждност изцяло; той позволява на приложението да определи кои данни изискват допълнителни гаранции и кои могат да бъдат изпратени без тях. Този подход е полезен и при други интерактивни системи, които обменят чести актуализации. Във всички случаи изборът на транспорт зависи от характера на данните, допустимото закъснение и поведението, което приложението трябва да поддържа при загуба на пакети. Развитието на интернет протоколите показва, че разграничението между TCP и UDP не трябва да се свежда до твърдението, че TCP се използва за надеждност, а UDP - само за приложения, при които загубата е приемлива. Съвременен пример е HTTP/3, който използва QUIC като транспортен протокол, работещ върху UDP [5, 6]. QUIC реализира функции, които не са част от самия UDP, включително надеждна доставка на потоци, криптографска защита и контрол на претоварването [4, 5]. Така приложението използва UDP за пренасяне на дейтаграмите, докато QUIC добавя необходимите механизми за управление на комуникацията. Тази архитектура позволява различен подход към установяването на защитена комуникация и управлението на множество потоци данни. Примерът с HTTP/3 демонстрира, че изборът на транспортна основа е част от по-широкото проектиране на протоколите. При оценката на дадена технология трябва да се разглеждат функциите на всички съответни слоеве, а не само характеристиките на UDP или TCP поотделно. В таблица 2 е показано сравнение на времето за предаване на данни чрез TCP и UDP, а фиг. 5 резултат от данните.

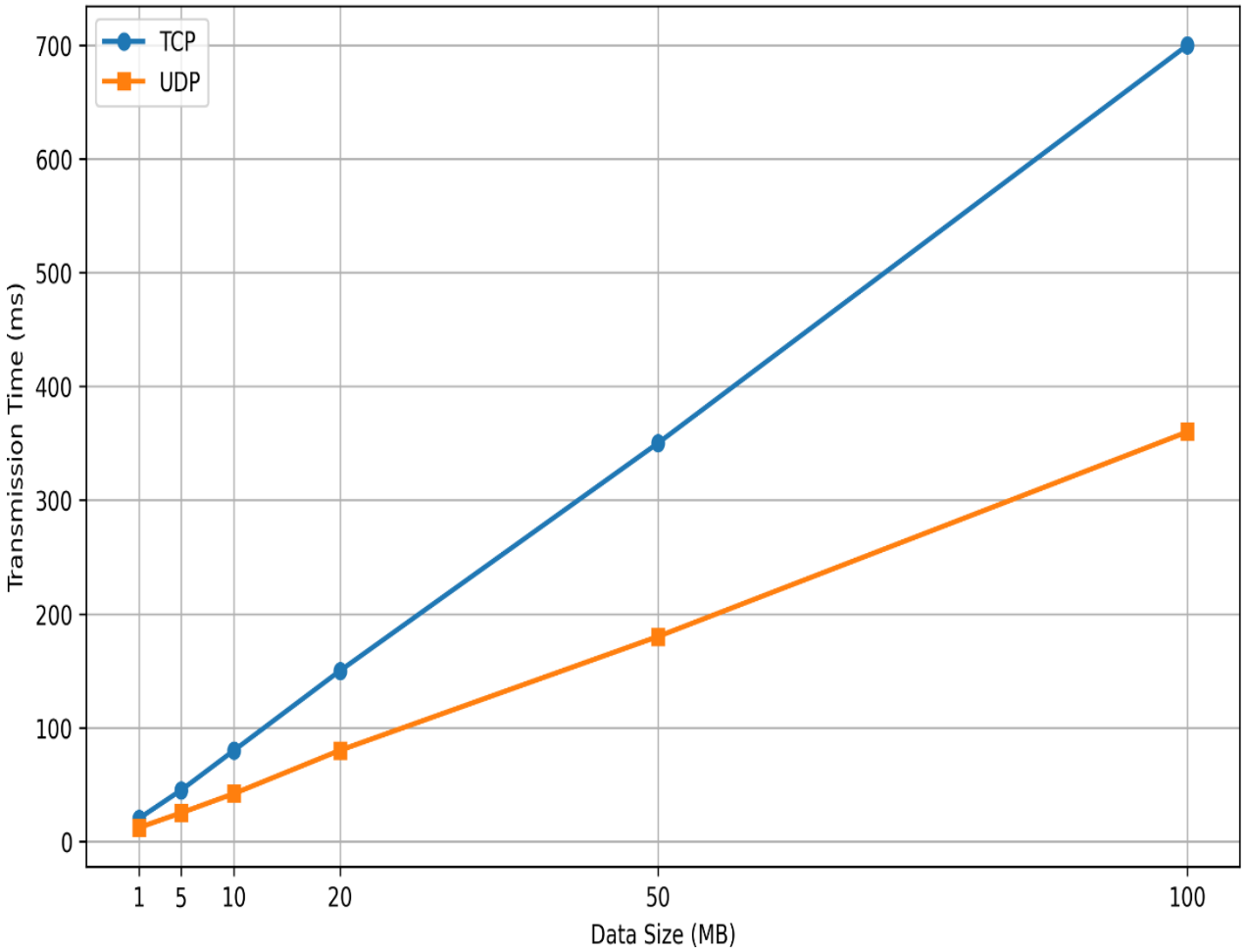
Таблица 2. Сравнение на времето за предаване на данни чрез TCP и UDP

Размер на данните (MB)	TCP (ms)	UDP (ms)
1	20	12
5	45	25
10	80	42
20	150	80
50	350	180
100	700	360

7 Заключение

Транспортните протоколи TCP и UDP имат съществено значение за функционирането на съвременните интернет комуникации. Те осигуряват обмен на данни между приложения, работещи на различни крайни устройства, но използват различни механизми за предаване и обработка на информацията. Разглеждането на техните характеристики показва, че изборът на транспортен протокол трябва да бъде съобразен с изискванията на конкретното приложение. TCP предоставя надежден и подреден поток от байтове чрез механизми за потвърждение, повторно предаване, контрол на потока и контрол на претоварването. Тези свойства го правят подходящ за услуги, при които е важно данните да бъдат получени правилно и в определен ред, като уеб комуникации чрез HTTP/1.1 и HTTP/2, отдалечено администриране чрез SSH и прехвърляне на файлове. UDP предлага по-опростен обмен на отделни дейтаграми, без предварително установяване на транспортна връзка и без вградени

гаранции за доставка и подредба. Този подход е полезен при кратки заявки и отговори, както и при приложения, чувствителни към времето, например някои DNS комуникации, гласови и видео приложения в реално време и определени видове игрови трафик. Когато са необходими допълнителни гаранции, те могат да бъдат реализирани от приложението или от протокол от по-високо ниво. Сравнителният анализ показва, че нито TCP, нито UDP е универсално по-добър във всички ситуации. TCP осигурява полезни механизми за надеждност, но повторното предаване и поддържането на връзка могат да повлияят върху закъснението. UDP има по-малко функции на транспортно ниво, но това не гарантира автоматично по-висока производителност. Реалното поведение зависи от мрежовите условия, вида на данните и изискванията на приложението. Примерът с QUIC и HTTP/3 допълнително показва, че съвременните комуникационни решения могат да комбинират UDP като транспортна основа с допълнителни механизми за надеждност, защита и контрол на претоварването. Това подчертава значението на взаимодействието между протоколите от различните слоеве на TCP/IP архитектурата. В заключение, познаването на принципите на работа, предимствата и ограниченията на TCP и UDP е важно за разбирането на интернет комуникациите и за правилния избор на транспортен механизъм. Изборът следва да се основава на необходимата надеждност, допустимото закъснение, обема на обменяните данни и начина, по който приложението обработва загуби и грешки.



Фиг. 5. Зависимост на времето за предаване от обема на данните при TCP и UDP

ЛИТЕРАТУРА:

[1] Стоянова, В., Талев, Д. Инструменти за анализ на трафика на локална мрежа. International Conference “Advanced Research and Technology for Defence” – ARTDef 2022, 2022, с. 133–141. ISSN 2815-2581.

[2] Eddy, W. (Ed.). Transmission Control Protocol (TCP). RFC 9293, Internet Engineering Task Force, 2022. https://www.rfc-editor.org/rfc/rfc9293

[3] Postel, J. User Datagram Protocol. RFC 768, Internet Engineering Task Force, 1980. https://www.rfc-editor.org/rfc/rfc768

[4] Thomson, M., Turner, S. Using TLS to Secure QUIC. RFC 9001, Internet Engineering Task Force, 2021. https://www.rfc-editor.org/rfc/rfc9001

[5] Iyengar, J., Thomson, M. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000, Internet Engineering Task Force, 2021. https://www.rfc-editor.org/rfc/rfc9000

[6] Bishop, M. (Ed.). HTTP/3. RFC 9114, Internet Engineering Task Force, 2022. https://www.rfc-editor.org/rfc/rfc9114