24. fejezet – A szállítási réteg

A szállítási réteg

A rétegek közül a szállítási réteg az alsó három réteg logikai folytatásának tekinthető, hiszen ha egy hoszt üzenetet küld a másiknak, akkor az üzenet továbbítása előtt ezt általában csomagokra kell darabolni, ezeket a hálózati rétegnek átadva át kell vinni a hálózaton és a célhosztnak átadni, ahol az üzenet összerakásra kerül. Az üzenetben leírt különféle típusú tevékenységet pedig végre kell hajtani.
A szállítási réteg feladata megbízható adatszállítás biztosítása a forráshoszt és a célhoszt között, függetlenül az alatta lévő rétegek kialakításától. A szállítási réteg feladata ellátásához a hálózati réteg által nyújtott szolgálatokra támaszkodik. A feladat itt már a tényleges hoszt-hoszt kapcsolat hibamentes megvalósítása. A szállítási réteg tehát már valódi két végpont között réteg, ami azt jelenti, hogy itt a forráshoszt és a célhoszt egymással kommunikál, míg az alsóbb rétegeknél ez nem igaz, ott a hosztok a szomszédjukkal (azaz azzal az aktív hálózati eszközzel, amihez közvetlenül kapcsolódnak) folytatnak párbeszédet. A használt protokollok sok esetben hasonlítanak az adatkapcsolati réteg protokolljaira, de itt az IMP-ket összekötő fizikai csatornát, a két hoszt között található teljes hálózat jelenti.
A szállítási réteg adategysége a szegmens (Segment), illetve használatos még a TPDU (Transport Protocol Data Unit / Szállítási Protokoll Adategység) elnevezés is. Az átvitel során a szegmensek, melyeket szállítási réteg használ csomagokba ágyazódnak, melyeket a hálózati réteg használ; a csomagok tartalma pedig keretekben folytatja útját az adatkapcsolati rétegben.
Az Internet két fő protokollt használ a szállítási rétegben, ezek az UDP és a TCP. Fontos megemlíteni továbbá a RTP protokollt, mely a valósidejű átvitelt biztosítja.

Az UDP protokoll

A felhasználói datagram protokoll (UDP – User Datagram Protocol) egy nem megbízható, összeköttetés nélküli protokoll. Az UDP protokoll leírását az RFC768 dokumentum tartalmazza. Elsősorban olyan egylövetű, kliens-szerver típusú kérdés-válasz alkalmazásokban terjedt el, ahol a gyors válasz sokkal fontosabb, mint a pontos, megbízható válasz, ugyanis az UDP nem garantálja a csomagok megérkezését. UDP-t használnak például az audio-video streaming alkalmazások, a DNS (Domain Name Server), az SNMP (Simple Network Management Protocol) és a DHCP (Dynamic Host Configuration Protocol) is.

Az UDP működési mechanizmusa a következő. Az UDP az információját egy IP csomagba helyezi el, ellenőrző összeget számol hozzá és feladja a csomagot. Így a kézbesítést nem garantálja, de a hibás kézbesítést észlelhetővé teszi. Ha a kérdés, vagy a válasz az átvitel során elveszik, a hiba egyszerű újrakérdezéssel megoldható, amennyiben az adott információ időben még releváns. Az UDP – mivel egy szállítási protokoll az jellemzően az operációs rendszer része – alkalmazásnak tekinthető. Az UDP egyszerűsége miatt kíméletesen bánik a hálózati erőforrásokkal.
Az UDP adatszerkezete is fejrészből valamint az adatokból áll. A fejrész 32 bites és mindössze 8 bájt hosszú, az adatok maximális méretét pedig az IPv4 csomag törzsrészének maximális hossza, mínusz az UDP fejrész hossza határozza meg, tehát 65515-8 = 65507 bájt.

← — — — — — — — — — — — — — — — — 32 bit — — — — — — — — — — — — — — — →

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Forrás port száma	Cél port száma
UDP szegmens hossza	UDP ellenőrző összeg
Adatok	

Az UDP adatszerkezetében egy új fogalom jelent meg, a port (Port), amit nagyritkán kapu néven is említ néhány forrás. A portok a hosztokon belül futó alkalmazások megkülönböztetésére szolgálnak. Ez azt jelenti, hogy bizonyos alkalmazások a számukra meghatározott portokat használják a kommunikációra.

Az UDP szegmens adatszerkezetének mezői a következők:

- Forrás port száma (Source Port)
  A forrás címe a forrás hosztot azonosítja, a forrás port száma pedig a forráson futó alkalmazást. Válaszküldés esetén ebből a forrás portból lesz a címport, azaz a forrás port ismerte nélkül nem lenne lehetőség válasz eljuttatására a forrás megfelelő alkalmazáshoz.
- Cél port száma (Destination Port)
  A cél hoszton futó alkalmazást azonosítja.
- UDP szegmens hossza (UDP Total Length)
  A szegmens hossza egyenlő a 8 bájtos fejrész és az adatmező együttes hosszával. A hossz értéke bájtonként értelmezendő, azaz 8 bájt (ebben az esetben csak a fejlécet tartalmazza, adattartalom nélkül) és 65515 bájt között lehet.

- UDP ellenőrző összeg (UDP Checksum)
  Az ellenőrző összeg használata nem kötelező, de növeli az adatbiztonságot.
- Adatok (Application Data / Message)

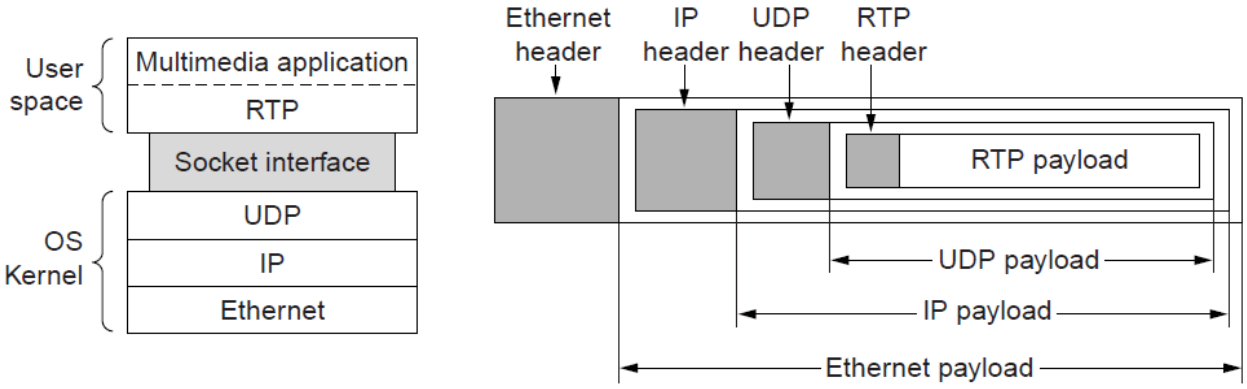
A portszámokat tartalmazó mezők, azaz a portok ismerete nélkül a szállítási réteg nem képes a szegmenseket a megfelelő alkalmazáshoz eljuttatni. Az, hogy az UDP a portokat már a fejrészben használja, jelentős idő- és teljesítmény előnyt jelent egy hagyományos IP csomag feldolgozhatóságához képest. A szegmensek IP csomagba történő beágyazása tehát megoldja az adatok megfelelő alkalmazáshoz történő gyors eljuttatását.

A port tartományokat eredetileg az RFC1700 dokumentum tartalmazta, de tekintettel a hatalmas tartomány szinte folyamatos változásaira (főleg a magasabb portszámok esetében), az aktuálisan hivatalos állapotot célszerű az Interneten megtekinteni itt: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt
A port tartományok rövid áttekintése:

- 1-1023 Közismert portok (Well-known) [csak a leggyakoribbak vannak felsorolva]
  - 20, 21 FTP állományátvitel
  - 22 SSH távoli bejelentkezés, telnet
  - 25 SMTP e-mail küldés
  - 53 DNS névszolgáltatás
  - 80 HTTP web elérés
  - 110 POP3 e-mail hozzáférés
  - 143 IMAP e-mail hozzáférés
  - 443 HTTPS biztonságos web elérés
  - 543 RTSP médialejátszó vezérlés
  - 631 IPP nyomtatómegosztás
- 1024-49151 Regisztrált portok (Registered)
  alkalmazás specifikus portok (NFS, BitTorrent, Antivirus, hálózatos játék, stb.)
- 49152-65535 Ideiglenes portok (Temporary / Dynamic)
  nem fixen kiosztott tartomány, kliens-szerver és ideiglenes egyedi alkalmazásokhoz

Az RTP protokoll

A valósidejű átviteli protokoll, az RTP (Real-time Transport Protocol) az UDP protokollra épülve, és így kihasználva annak multiplexelő és hibadetektáló funkcióit – a szegmensek valósidejű továbbítását biztosító megoldás. Az RTP protokoll leírását az RFC1889 dokumentum tartalmazza. Az RTP protokoll nem közvetlenül foglalkozik a szegmensek gyors továbbításával, vagy QoS biztosításával, ezeket az alatta levő rétegekre bízza. Az RTP feladata csupán a valósidejű kommunikáció menedzsmentje.
Az RTP adatfolyamokat nem az operációs rendszer, hanem maga a valósidejű alkalmazás darabolja szegmensekre, tehát az RTP nem az operációs rendszer része, hanem az alkalmazásoké. A szegmensek belső formátuma a fejléctől eltekintve az alkalmazásra van bízva, így maga az RTP csak egy keretprotokoll, konkrét alkalmazásához ki kell egészíteni a használt szegmensek és csomagok típusával, a típuskódok és az egyes típusú adatok kódolásának leírásával. Mindezekkel az RTP nem foglalkozik, csupán a szegmensek átvitelével, a szinkronizációs információ előállításával és kezelésével, valamint a kapcsolat minőségének felügyelésével. Az RTP helyzetét az egymásra épülő protokollokban az ábra szemlélteti.



Az RTP fejrésze 32 bites. Fixen 3 db 32 bites szóból, azaz 12 bájtból, és néhány kiterjesztésből áll.

← — — — — — — — — — — — — — — — — 32 bit — — — — — — — — — — — — — — — — →

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Verzió		P	X	CC				M	Adatmező típusa							Sorszám															
Időbélyeg																															
Szinkronizációs forrás azonosítója (SSRC)																															
Közreműködő források azonosítója (CSRC)																															

Az RTP fejrészének mezői a következők:

- Verzió (Version)
  Ez a mező csak 2 bites, értéke jelenleg 2, azaz további két verziót képes jelezni.
- P (Padding)
  Értéke 1, ha maga a TPDU nincs teljesen kitöltve adattartalommal, azaz a formai követelmények miatt ki kellett egészíteni 4 bájtra, vagy annak valamelyik többszörösére. Az utolsó, a kiegészítő bájt tartalmazza, hogy hány bájtot kell figyelmen kívül hagyni, azaz hogy mennyi volt a töltelék, a kitöltés.
- X (Extension)
  Értéke 1, ha a fejrész kiegészítő fejrésszel rendelkezik.
- CC (CSRC Count)
  Az CSRC azonosítók számát tartalmazza.
- M (Marker)
  Alkalmazásfüggő jelölőbit, egy TPDU folyam szignifikáns eseményeit jelölheti, például a kép- vagy hangkeret elejét vagy végét.
- Adatmező típusa (Payload Type)
  A használt kódolási algoritmus azonosítója. Például: mp3, mkv, stb.
- Sorszám (Sequence Number)
  A TPDU-k számolására szolgál, segítségével detektálható az adatvesztés.
- Időbélyeg (Timestamp)
  A valósidejű átvitelt segítő mező. Értéke a vevő oldali időzítési szórás (Jitter) csökkentésével segíti elő a multimédiás tartalom – csomagok érkezési idejétől független – folyamatos lejátszását.
- SSRC (Synchronization Source Identifier)
  A szinkronizációs forrás azonosítója az RTP folyam forrását azonosítja.
- CSRC (Contributing Source Identifier)
  Az RTP által létrehozott kombinált (mixelt) folyam közreműködő komponenseit azonosítja.

Az RTP protokoll szerepe tehát leginkább a valósidejű multimédia információk továbbításának végrehajtásában van, UDP protokoll igénybe vételével, az IP protokollon, azaz jellemzően az Interneten.

A TCP protokoll

Az átvitelvezérlő protokoll (TCP – Transmission Control Protocol), egy megbízható összeköttetés alapú protokoll. A TCP protokoll első verziójának a leírását 1981-ből az RFC793 dokumentum tartalmazza. Evolúciója során az RFC1122 (hibajavítások), az RFC1323 (teljesítőképesség növelése), az RFC2018 (szelektív nyugtázás), az RFC2581 (torlódáskezelés), az RFC2873 (fejrész továbbfejlesztése), az RFC2988 (időzítők), az RFC3168 (explicit torlódáskezelés) és az RFC4614 (a jelenlegi verzió) dokumentumok is a TCP-vel foglalkoztak. Felépítése jóval bonyolultabb, mint az UDP felépítése. Feladata az, hogy hibamentes bájtos átvitelt biztosítson bármely két hoszt között az Interneten, az egyébként megbízhatatlan összekapcsolt hálózaton – ezért ellenállónak kell lennie a lehetséges meghibásodásokkal szemben. A TCP átvitele tehát egy bájtfolyam, a rendszer az üzenethatárokkal nem foglalkozik. A TCP forgalomszabályozást is végez annak érdekében, hogy egy gyors forráshoszt csak annyi üzenetet küldjön egy lassabb célhosztnak, amennyit az fogadni képes. A TCP mivel egy szállítási protokoll az jellemzően az operációs rendszer, azon belül is leggyakrabban a kernel része.
A TCP-t használó hosztok az adatokat szegmensekben cserélik egymás között. A szegmensméret korlátja csak az, hogy a TCP fejrésznek és a szegmensnek egyaránt be kell férnie az IPv4 csomag törzsrészébe. A szegmens maximális hossza az IPv4 törzsrész mínusz a TCP fejrész hossza, azaz 65515-20 = 65495 bájt.

A TCP fejrésze 32 bites. Fixen 5 db 32 bites szóból, azaz 20 bájtból, és néhány opcionális mezőből áll.

← — — — — — — — — — — — — — — — — 32 bit — — — — — — — — — — — — — — — — →

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Forrás port száma																Cél port száma															
Sorszám																															
Nyugtaszám																															
TPC fejrész hossza								CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Ablakméret															
TCP ellenőrző összeg																Sürgősségi mutató															
Opciók (0 vagy több 32 bites szó)																															
Adatok																															

A TCP szegmens adatszerkezetének mezői a következők:

- Forrás port száma (Source Port)
  A forrás címe a forrás hosztot azonosítja, a forrás port száma pedig a forráson futó alkalmazást. Válaszküldés esetén ebből a forrás portból lesz a címport, azaz a forrás port ismerte nélkül nem lenne lehetőség válasz eljuttatására a forrás megfelelő alkalmazáshoz.
- Cél port száma (Destination Port)
  A cél hoszton futó alkalmazást azonosítja.
- Sorszám (Sequence Number)
  A mező tartalma egy 32 bites, átforduló szám, amelynek a kezdeti, véletlenszerűen választott sorszámértékhez képesti eltolása megegyezik az adott szegmens első adatbájtjának az bájtfolyamban elfoglalt pozíciójával. A vett szegmens sorszáma alapján dönt a vevő arról, hogy ez beleillik-e a vételi ablakába vagy el kell-e dobnia. A 32 bittel nagyjából 4 GB adat címezhető meg.
- Nyugtaszám (Acknowledgement Number)
  A mező azt a következő adatbájt sorszámát tartalmazza, amire a hoszt éppen várakozik. Az ACK flag azt jelzi, hogy ez a mező érvényes adatot tartalmaz-e.
- TCP fejrész hossza (TCP Header Length)
  Azt mutatja meg, hogy a TCP fejrész hány darab 32 bites szót tartalmaz. Az opció mező változó hossza miatt, erre az információra feltétlenül szükség van.
- Használaton kívüli 4 bit (Not In Use)
- 8 db egybites jelző mező (Flag)
  A jelzőbitek a TCP kapcsolat adataira, állapotaira, állapotváltozásaira vonatkozó kérésekben és jelzésében, illetve a torlódáskezelésben játszanak szerepet.
    - CWR (Congestion Windows Reduced)
    - ECE (ECN Echo)
    - URG (Urgent Pointer)
    - ACK (Acknowledgment)
    - PSH (Push Function)
    - RST (Reset the Connection)
    - SYN (Syncronize Sequence Number)
    - FIN (Final, No more Data from the Sender)
- Ablakméret (Window)
  A forgalomszabályzásnál használt változó méretű csúszóablak méretét jelzi.
- TCP ellenőrző összeg (Checksum)
  Az UDP-vel ellentétben az ellenőrző összeg használata TCP esetén kötelező.

- Sürgősségi mutató (Urgent Pointer)
  Ha az URG flag értéke 1, akkor ez a 16 bit az aktuális sorszámtól számolva kijelöli a sürgős adatok kezdetét.
- Opciók (Options)
  Amennyiben a hosztokon futó alkalmazások igénylik, a TCP esetében lehetőség van opciók egyedi használatára. Leggyakrabban a puffer méret egyeztetésre használják a hosztok vonatkozó alkalmazásai.
- Adatok (Data)

A portok szerepről az UDP protokoll tárgyalásakor már volt szó. Annak érdekében, hogy a kapcsolatban részt vevő alkalmazást azonosítani lehessen, illetve, hogy egy hoszt egyszerre több élő TCP kapcsolattal rendelkezhessen, a TCP adatot hordozó IP csomagokban nemcsak a célhoszt címet kell megadni, hanem a TCP port számát is. A TCP összeköttetés duplex pont-pont összeköttetés, azaz a forgalom egyszerre két irányba halad, és az adatszórás (Broadcast) valamint a többesküldés (Multicast) nem támogatott.
A tehát TCP egy kapcsolat-orientált protokoll, fő feladata egy megbízható, és biztonságos kapcsolat kiépítése és fenntartása két folyamat között. A megvalósítás menetét alapvetően három részre bonthatjuk:

1. Létrejön a megbízható kapcsolat két állomás között.
2. Megkezdődik a tényleges adatátvitel.
3. A kapcsolat lezárása, és a számára elkülönített erőforrások felszabadítása.

A protokoll a hibamentes átvitelhez az úgynevezett pozitív nyugtázás újraküldéssel (Positive Acknowledgement with Retransmission) eljárást használja. A TCP kapcsolatok egyes lépéseit állapotoknak nevezzük. A kapcsolat az élettartama alatt különböző állapotváltozásokon megy keresztül:

- CLOSED
  Ez az alapértelmezett állapot, amelyből a kapcsolat kiépítésének folyamata indul. Elméleti állapot, a hosztok között nincs élő, létező kapcsolat (vagyis még nem jött létre, vagy már lezárult).
- LISTEN
  A hoszt (általában a szerver) szinkronizálási kérésre várakozik (SYN), saját SYN üzenetét még nem küldte el.
- SYN-SENT
  A hoszt (általában a kliens) elküldte a SYN üzenetet, és várakozik a válaszra a másik hoszttól (általában a szervertől).

- SYN-RECEIVED
  Kapcsolódási kérés (SYN) elküldve és fogadva is, várakozás a másik hoszt általi nyugtázás beérkezésére (ACK).
- ESTABLISHED
  A létrejött TCP kapcsolat stabil állapota. Miután mindkét hoszt ebbe az állapotba kerül, megkezdődhet az adatok átvitele, ami addig folytatódhat, amíg valamelyik hoszt a kapcsolat lezárását nem kezdeményezi.
- CLOSE-WAIT
  Az egyik hoszt kapcsolatbontási kérést (FIN) kapott a másik hoszttól. Várakozik a helyi alkalmazás nyugtázására, mielőtt elküldené a megfelelő válaszüzenetet.
- LAST-ACK
  A hoszt már fogadott és nyugtázott egy kapcsolatbontási kérést, el is küldte a saját FIN üzenetét, és várakozik a másik hoszt ezen kérésre érkező nyugtájára (ACK).
- FIN-WAIT-1
  Várakozás az elküldött FIN üzenet nyugtázására, vagy a kapcsolatbontási kérés érkezésére másik hoszttól.
- FIN-WAIT-2
  Megérkezett a nyugta az elküldött kapcsolatbontási üzenetre, várakozás a másik hoszt FIN üzenetére.
- CLOSING
  A hoszt megkapta a másik hoszt FIN üzenetét, és nyugtázta azt, de a saját FIN üzenetére nyugtát még nem kapott.
- TIME-WAIT
  A kapcsolatbontási kérést és a nyugtát (FIN, ACK) a hoszt megkapta és kiküldte, a kapcsolat lezárult. Egy rövid ideig várakozik még, hogy biztosítsa azt, hogy a másik hoszt is megkapja a nyugtát, és hogy ne legyen átfedés az újonnan létrejövő kapcsolatokkal.