26. fejezet

Számítógépes hálózatok és internet-eszközök

Fizikai réteg, adatkapcsolati réteg, hálózati réteg, szállítói réteg – feladatok, módszerek, protokollok.

26.1. Bevezetés

26.1.1. Hálózatok fajtái

LAN = *Local Area Network*: helyi hálózat, kis körzetben (egy épületben, kampuszon) elhelyezkedő számítógépeket köt össze,

MAN = *Metropolitan Area Network*: lakónegyedet lefedő (TV-kábel alapú) hálózat,

WAN = *Wide Area Network*: nagyobb területeket (kerület, város) átfogó hálózat,

Internet (nagyjából) az egész Földre kiterjedő WAN.

26.1.2. Topológiák

A különböző kommunikációs hálózatok más más topológiákat (összeköttetési elrendezéseket) használnak.

LAN topológiák:

- gyűrű,

- busz,
- csillag.

WAN-topológiák:

- központosított,
- központosítatlan.

Internet: alhálózatok (subnet) adják a helyi hálózatok kapcsolatát.

Átvitel fajtái.

Unicast: pontosan két résztvevő kommunikációja (pl. telefon),

Multicast: egy feladó egyszerre több címzett számára küld üzenetet (pl. videokonferencia),

Broadcast: a feladó az összes elérhető fogadó felé továbbítja a jeleket (pl. rádió).

26.1.3. Történetének néhány állomása

1969 – ARPANET Larry Roberts által az UCLA-n (Los Angelesben) szervezett kis, helyi hálózat.

1972 Robert Kahn koncepciója a „legjobb szándékról", illetve a feketedoboz-kapcsolatokról (önálló helyi hálózatok, szabványos kapcsolatok kifelé) és a folyamfelügyelet elhaygásáról.

1988 – NSFNET tudományos hálózat az USA-ban (egyik végétől a másikig).

26.1.4. Rétegek, protokollok

A kommunikációs protokollok réteges felépítését a nagyszámú különböző adattovábbító eszköz és felhasználási mód hívta létre. Köztes, szabványos rétegek bevezetésével jelentősen csökkennek a fejlesztési és karbantartási költségek, követhetővé válik a fejlődés.

A TCP/IP rétegmodell.

- felhasználói,
- szállítói,
- hálózati,
- adatkapcsolati.

Az ISO rétegmodell.

- felhasználói (alkalmazások szintje),
- prezentációs (az adatok ábrázolásának egységes kezelése),
- viszony */session/* (kapcsolatok felépítése, bontása, kapcsolat jellegének meghatározása),
- szállítói (adat csomagokra bontása, folyamfelügyelet),
- hálózati (csomagtovábbítás, útvonalmeghatározás),
- adatkapcsolati (átviteli hibák szűrése, közös átviteli médiumok kezelése, nyugtázás),
- fizikai (bitek továbbítása, elektronikai megoldások).

Hibrid modell (Tannenbaum).

- felhasználói,
- szállítói,
- hálózati,
- adatkapcsolati,
- fizikai.

26.2. Fizikai réteg

A fizikai réteg az adatok jelekké való leképezésével, illetve azok továbbításával foglalkozik.

26.2.1. Kódolások

26.2.1. Definíció (Baud). *Az adatátvitel alapegysége a szimbólum. Az adatátviteli sebesség mértéke lehet a szimbólumráta (egysége: Baud = szimbólum/másodperc).*

A fizikai réteg első kérdése a bináris információ elektronikus reprezentációja. Ennek legegyszerűbb módja a bitek leképezése egyszerű feszültségváltakozásra – ez azonban nem hatékony.

Szinkronizáció

A szinkronizáció problémája azt jelenti, hogy ha két, eltérő vagy egymástól elcsúszó órajellel rendelkezik a küldő és a fogadó, akkor az érkező elektromos/optikai jelsorozatot nem megfelelően értelmezi.

Megoldási ötletek:

1. explicit órajel: az órajelet külön csatornán szinkronizáljuk,
2. szinkronizálás kritikus időpontokban: ha bízhatunk abban, hogy az órák rövid idő alatt nem tolódnak el kritikusan, akkor csak pl. minden adatblokk elején kell szinkronizálni,
3. órajel kikövetkeztetése a szimbólumok kódolásából.

Különböző digitális kódok.

- NRZ-L (1 = magas, 0 = alacsony),
- NRZ-M (1 = váltás az idő elején, 0 = nincs váltás),
- NRZ-S (NRZ-M fordítottja),
- RZ (1 = magas-alacsony, 0 =alacsony),
- Biphase-level Manchaster (1 = magas-alacsony, 0 = alacsony-magas),
- Biphase-Mark (minden idő elején váltás, illetve 1 esetén a közepén is),
- Biphase-Space (fordítva, mint a Biphase-Mark),
- Differential Manchaster (minden intervallum közepén váltás, 0 esetén az elején is),

- Delay Modulation (1 = váltás középen, 0 = váltás a végén, ha 0 következik, nincs váltás, ha 1 következik).

Az utolsó 5 kódolás önütemező, tehát az órajel kikövetkeztethető a szimbólumok kódolásából.

Jelölések.

N : Non

R : Return-to

Z : Zero

L : Level

M : Mark

S : Space

Alapsáv és szélessáv

Alapsávú átvitelnél a jel közvetlenül árammá alakul, és minden frekvencián egyszerre kerül átvitelre. A *szélessávú* átvitel segítségével nagyobb átviteli sebességet érhetünk el, mivel az adatokat csak egy meghatározott frekvenciatartományon visszük át.

Vivőhullámok és felhasználásuk. A szélessáv vivőhulláma valamilyen szinuszgörbe (e nem hordoz információt), ezzel kombináljuk a jelsorozatot.

Amplitúdó-moduláció: a jel változását a szinuszgörbe amplitúdója írja le. Az egyes jelekhez (szignálokhoz) különböző amplitúdók tartozhatnak.

Frekvencia-moduláció: a vivőhullám frekvenciáját változtatjuk a jel függvényében. Az egyes szimbólumokhoz tehát különböző frekvenciákat rendelünk.

Fázis-moduláció: a vivőhullám fázisát eltoljuk a jelnek megfelelően.

Kombinált modulációk: nagyobb jelkészlet továbbítható, ha egyszerre többféle modulációt alkalmazunk.

Bithiba-gyakoriság

26.2.2. Definíció (BER (bit error rate). *A hibásan fogadott bitek aránya a fogadott bitekhez viszonyítva.*

A hibagyakoriság erősen függ a szignál-zaj aránytól, ezen keresztül az átviteli sebességtől, médiumtól, annak zajérzékenységétől, a szignál erősségétől.

26.2.2. Fizikai médiumok

Vezetékes átvitel

UTP (Unshielded twisted pair). Két kategóriája elterjedt: az 5. kategóriás UTP sűrűbben sodort, emiatt kevésbé zajérzékeny,mint a 3. kategóriás.

Koaxiális kábel. Árnyékolt, erősen szigetelt kábeltípus.

Optikai szál. Az üveg határfelületén jelentkező teljes visszaverődés jelenségét használja ki a fénnyé alakított jel továbbítására. Két különböző törésmutatójú üvegszálat helyezünk el egymás mellett. Sebességének fő korlátja a végpontokon található átalakító/érzékelő eszköz.

Vezeték nélküli átvitel

Infravörös fény. Kis hatósugarú, könnyen elnyelődik. gyakran használják távirányítókban és hasonló eszközökben.

Rádióhullámok. Sokféle frekvenciatartományt különböző célokra használnak. A nagyobb frekvenciák könnyebben elnyelődnek, egyenesebb vonalon terjednek és visszaverődhetnek az ionoszféráról. A kisebb frekvenciák jobban követik a Föld görbületét.

Problémák.

- több utas terjedés (elhajlás, visszaverődés),
- szakadások,
- vételi erősség változása.

Multiplexelés (médiumok közös használata)

Tér-multiplexelés. A rádióhullámok távolságból adódó jelgyengülése kihasználható úgy, hogy különböző, távol eső területeken ugyanazt a frekvenciát más adatok átvitelére használhatjuk probléma nélkül.

Térmultiplexelés valósítható meg irányított antennák használatával is.

Frekvencia-multiplexelés. A hullámsávot frekvenciatartományokra osztva az egyes tartományokon más-más jelet továbbíthatunk.

Idő-multiplexelés. A legegyszerűbb módszer – a csatorna időbeli felosztása. Helyes működéséhez a résztvevők szinkronizációja, illetve valamilyen felosztási megegyezés/vezérlés szükséges.

Kód-multiplexelés. Egyik módja a CDMA (Code Division Multiple Access). Lényege, hogy a résztvevők egymásra ortogonális ún. *chip-kóddal* szorozzák az átvitt adatokat.

26.3. Adatkapcsolati réteg

Az adatkapcsolati réteg dolga, hogy a fizikai réteg bizonytalanságait a hálózati réteg felé elrejtse, és az adatoknak struktúrát biztosítson.

Feladatok.

1. a hálózati rétegtől kapott adatcsomagok keretekre bontása,
2. hibafelügyelet,
3. folyamfelügyelet.

26.3.1. Keretezés

A keretezés komoly problémája a kerethatárok megállapítása, figyelembe véve azt is, hogy a fogadó fogadhat biteket (pl. zaj) akkor is, ha a küldő nem küld semmit.

Kerethatárok hossz alapján. Ekkor a keretek fejlécében található a keret hossza. Probléma: ha a kerethossz hibásan kerül átvitelre, nem vesszük észre.

Fej- és láblécek. Speciális bitsorozatokkal látjuk el a keret elejét és végét. Elvárjuk, hogy ezek ne szerepeljenek a kereten belül. Ha mégis ez lenne a helyzet, *bájtbeszúrás*t (*escape*) alkalmazunk.

A módszer speciális változata a bitbeszúrás: a keretek végét jelezze adott hosszúságú (pl. 6 bit) 1-es! Ekkor a keretek belsejében minden 5 1-es bit után szúrjunk be egy 0-t (akkor is, ha utána 0 következik)!

Keretezés kódmegsértéssel. A fizikai réteg kódjának hibája jelzi a kerethatárokat. A módszer elterjedt (pl. ethernet), mert robosztus és olcsó.

26.3.2. Hibafelügyelet

A hibafelügyelet alapegysége a *keret* (*frame*) – tehát az adatkapcsolati réteg azt vizsgálja, hogy egy keret helyesen vagy hibásan érkezett meg.

Kétféle megközelítés. *Előre javítás* esetén a keretek redundáns kódolása segítségével a hibák fogadás utáni kijavítása; *utólagos javítás*kor a hiba felismerése után a keretet újrakérjük a küldőtől. Esetenként bizonyos mennyiségű hiba tolerálható (pl. hangátvitel).

Fogalmak:

- Hamming-távolság,
- kód,
- redundancia,
- kódráta: $R_S = \frac{\log |S|}{n}$,
- kódtávolság: $\delta_S = \frac{d(S)}{n}$.

Módszerek

Paritásbitek. Ld. a kódoláselméletről szóló 9. fejezetet!

CRC. Hatékony hibafelismerést biztosít, de javításra nem alkalmas. Ld. a kódoláselméletről szóló 9. fejezetet!

26.3.3. Folyamfelügyelet

Lényege a küldő és a fogadó adatátviteli sebességének egymáshoz igazítása, hogy a küldő ne árassza el felesleges, fel nem dolgozható adatokkal a fogadót.

Egyszerű szimplex-protokoll nyugtákkal. Várakozási időkorlátokkal kiegészítve már működik, de lassú lehet.

Duplikátumok kezelése. Ha a nyugta elveszett, a küldő újra fogja küldeni a már egyszer megérkezett csomagot – a duplikátumok kiszűrésére a csomagokat (és a hozzájuk tartozó nyugtákat) sorszámozzuk. Fontos, hogy két különböző, de azonos sorszámú csomag nem haladhat egyszerre!

Alternáló bit protokoll. A szimplex protokoll kiegészítése 0 és 1 sorszámokkal. Hatékonysága az időeltolódás miatt kicsi.

Csúszó ablak protokoll. A szimplex protokoll kiterjesztése több csomagra: a küldő több (n – ablakméret) csomagot küld egymás után, és ha az elsőre nyugta érkezett, akkor küldi az $n+1$-ediket.

Go back N. Ha a fogadó ablakmérete 1, akkor amíg nem fogadott egy adott csomagot, addig a rákövetkezőket sem. Tehát ha a várakozási idő lejár a küldőnél, akkor újra kell küldenie minden addig elküldött csomagot.

Szelektív ismétlés. Ha az ablakméret 1-nél nagyobb, akkor néhány csomagot pufferelhet.

Két irányú kommunikáció és hátizsák-technika. A nyugták és az adatcsomagok elkülönítve mindkét irányban haladnak vagy az – általánosabban elterjedt – hátizsák-technikát alkalmazzuk: a nyugtát az ellentétes irányba küldött adat-frame fejlécébe tesszük (*piggybacking*).

26.3.4. Médiumhozzáférési alréteg (MAC)

A médiumhozzáférési alréteg szabályozza a csatorna megosztásának módját a kommunikáció résztvevői között.

A MAC protokolloknak a *löketszerűen érkező adatok* problémájával kell elsősorban megküzdeni. ez azt jelenti, hogy a hálózat adatforgalma általában nem az átlagérték körül mozog, hanem általában alacsony, és néha nagyon magas (*löket*, *burst*).

Statikus multiplexelés

Lényege, hogy minden résztvevőhöz fix időegységeket/frekvenciákat,/csatornákat rendelünk. A módszer csak akkor hatékony, ha előre ismerjük az állomások adatrátáit, és ez alapján oszthatjuk el a csatornahasználatot (pl. telefonhálózat). Löketszerűen érkező adatok esetén nem hatékony.

Dinamikus multiplexelés

Dinamikus multiplexelés esetén a résztvevők között valamilyen módon „igény szerint" osztjuk el a csatornát. A dinamikus multiplexelés alapfogalma a *kollízió* (ütközés): a csatornán egyszerre csak egy csomag vihető át hibamentesen; illetve a *vivő-érzékelés*: vagyis hogy egy állomás meg tudja-e állapítani, hogy épp foglalt-e a csatorna.

Fontos mutatói:

- átvitel (csomagok száma időegység alatt),
- késés (a csomagok átlagos átviteli ideje),
- igazságosság (az állomásokra vetítve az átvitel és a késés egyenlő elosztása).

ALOHA (kollízió alapú). Ha egy csomag kész, azonnal kerüljön átvitelre! Az ütközések kezelésére csak nyugtákat használunk – ha a nyugta nem érkezik meg, újraküldjük a csomagot.

A módszer előnye az egyszerű kivitelezés, hátránya, hogy a nyugták feltétlenül szükségesek, továbbá nem mindig jó a csatorna kihasználása (ha sok a küldő, így az ütközés).

Slotted ALOHA (kollízió alapú). Legyenek az állomások valamilyen módon szinkronizálva! Ha egy csomag kész, mindig csak egy időegység elején lehet elküldeni. Így a csomagok „sebezhetőségi ideje" lecsökken.

Ez a megoldás nagy terhelésnél még mindig nem elég hatékony.

CSMA (Carrier Sense Multiple Access – kollízió alapú). Vivőérzékelést alkalmazó protokoll – csak akkor küld adatot, ha szabadnak látja a csatornát. Ebben azonban „tévedhet" az adatok terjedési sebessége miatt.

Legyen $0 \leq p \leq 1$ a CSMA *perzisztenciája*!

1. Ha a csatorna szabad, akkor

- p valószínűséggel küldjük el a csomagot, és ha ütközés van, várjunk véletlen ideig, majd menjünk 1-re,
- $1 - p$ valószínűségge várjunk a következő időszeletre, és menjünk 1-re.

2. Ha a csatorna foglalt, figyeljük, és ha szabaddá válik, menjünk 1-re.

Van még *1-perzisztens* és *nem perzisztens* CSMA is.

CSMA/CD. Ütközés felismerésekor az átvitelt megszakítjuk, és véletlen ideig várunk a következő próbálkozásra.

A véletlen várakozási időt választhatjuk az ütközések függvényében: ha történt egy ütközés, duplázzuk meg az intervallumot, amiből a véletlen várakozást választjuk! Ha az intervallum elért egy felső korlátot, csökkentsük vissza a kezdőértékre! Ez a *binary exponental backoff* módszer.

Bitmap-protokoll (versenymentes). Minden küldési periódus előtt mindenki jelzi, hogy akar-e küldeni: ha igen, kap időszeletet a küldési periódusban. Alacsony terhelés esetén a késés nagyobb, de nagy terhelésre az átvitel stabil, megbízható.

Adaptív fa (korlátozott versennyel). Ha egy időszeletben ütközés történt, akkor a következő időszeletben a résztvevőknek csak egy része próbálkozhat újra ... ezt folytatjuk, amíg nem sikerült az átvitel.

26.3.5. Az ehternet (IEEE 802.3)

Fizikai réteg: Manchaster kód ($\pm 0.85 Volt$),

Adatkapcsolati réteg: a kerethatárokat kódmegsértés jelzi, CSMA/CD (megfelelő minimális csomagmérettel: 1500 byte).

26.4. Hálózati réteg

A hálózati réteg feladatai:

- útvonalinformációk kezelése,
- a csomagok megfelelő útvonalra irányítása.

26.4.1. Internet Protocoll (IP)

A csomagtovábbítás *routing táblák* alapján történik: az elosztóállomások bizonyos címekhez vagy címcsoportokhoz (subnet) ismerik a következő állomás (*gateway*) címét – ide továbbítják a csomagot. Ha az átjáró nem ismeri az útvonalat a célhoz, akkor az alapértelmezett átjárón (*default gateway*) továbbítja azt.

Minden IP-csomag rendelkezik egy TTL (*Time To Live*) értékkel – ha ennyi ugrással nem találja meg a célt, akkor eldobják.

A csomagtovábbítás menete. Ha a routerbe egy csomag érkezik, akkor:

- TTL := TTL - 1,
- Ha TTL $>$ 0, akkor:
  - továbbítjuk a csomagot a routing tábla alapján,
- különben ha a csomag nem ICMP-csomag (Internet Control Message Protocol), akkor:
  - a feladónak küldjünk ICMP-csomagot a router IP-címével, mint feladóval, melyben jelezzük, hogy a TTL „elfogyott”.

Statikus routing. A routing táblát kézzel építjük fel. Nyilván csak kis hálózatokon hatékony (ott jó is lehet).

Dinamikus routing

Distance Vector módszer. A Bellman-Ford algoritmus „elosztott” változatát használja, ahol minden router csak a szomszédaival kommunikál. A routing tábla tartalmazza a továbbítás becsült költségét és a következő állomás számát minden lehetséges célhoz.

Probléma: count to infinity (egy állomás kiesése csak lassan derül ki). Részelges megoldások: *split horizon (with poison reverse)* – A nem küldi vissza B-nek, amit „tőle tanult”.

Link State protokoll. Nem elosztott algoritmus, melyben minden router tárolja a hálózat teljes topológiáját.

Hierarchikus routing

A nagy routing-táblák kiküszöbölése – automóm rendszerek (Autonomous System, AS), inter-AS és intra-AS routing.

Multicasting

Egyszerű megoldás lenne minden címzettnek elküldeni egy önálló csomagot, de ez nagy terhelést jelent – általában a routerek döntik el, hogy duplikálnak-e a mögöttük álló állomások felé.

DHCP (Dynamic Host Control Protocol)

IP-címek automatikus kiosztása – ha egy gép csatlakozik, a DHCP-szervertől kér egy IP-címet.

26.5. Szállítói réteg

A szállítói réteg feladatai:

- interakciós modell nyújtása (pl. kapcsolatorientált/kapcsolat nélküli),
- megbízhatóság meghatározása,
- hibafelügyelet magasabb szinten,
- torlódás-feügyelet,
- adatok továbbítása a felhasználói réteg megfelelő alkalmazásának (pl. böngésző).

26.5.1. TCP (Transmission Control Protocol)

Kapcsolatorientált, megbízható bájtfolyamot biztosító protokoll – tehát minden csomagot nyugtáz, és külön ellenőrző összeget helyez a csomagokra. Többszörös küldés nem támogatott.

A fejléc felépítése.

- forrás- és célport,
- opciók (flagek, pl. ACK),

- sorszám,
- nyugtaszám (az első, még fel nem dolgozott, de fogadott byte sorszáma – piggybacking),
- ellenőrző összeg.

Kapcsolatfelépítés és -lezárás. Háromutas kézfogással épül fel a kapcsolat. A lezárásban 4 csomag vesz részt (két félig-lezárás).

Torlódás-felügyelet, csatornakihasználás

Nagle. A kisebb csomagok addig nem kerülnek kiküldésre, amíg nincs meg minden nyugta. Ha egy nyugta megjött, küldi a következőt.

Csúszó ablak. ld. fentebb.

Slow Start. A küldő kis értékről indítva folyamatosan növeli az ablakméretét, amíg egy nyugta ki nem marad.

26.5.2. UDP (User Datagram Protocol)

Kapcsolat nélküli protokoll. A fejléc csak a küldő és a fogadó portszámot, illetve az adat hosszát és ellenőrző összegét hordozza.

26.6. Felhasználói réteg

26.6.1. DNS

Feladata az IP-címek leképezése az ember számára könnyebben kezelhető nevekre. Ezt a DNS-szerverek közötti elosztott adatbázis segítségével teszi.

A nevek a pontok mentén hierarchikusan strukturáltak. Egy domain (pl. hu) alatti névteret a domain tulajdonosa felügyeli saját szervereivel. A *root* zóna felügyelete 13 szerver kezében.

A feloldás történhet iteratívan (a megkérdezett szerver pontosan annyi információt ad vissza, amennyit ő tud) vagy rekurzív (a névszerver a többi névszerver segítségével kiderít mindent, ami kell, és ezt adja vissza – jellemzően lokális névszervereknél). A kommunikáció UDP protokollt használ.

Gyors válaszismétlés: a visszaadott válaszok tárolódnak a szerveren.

Dinamikus DNS. Dinamikus címkiosztás esetén a DHCP-szerver tájékoztatja a közeli DNS-szervert az új csatlakozóról (rövid TTL-lel).

Reverse Name Lookup. 9.161.181.157.in-addr.arpa

26.6.2. E-mail

Résztvevői: *user agent* és *message transfer agent*. Szolgáltatások: küldés, értesítés, megjelenítés, válasz, továbbküldés, automatikus válasz, levelező listák, ... Protokollok: SMTP/IMAP és POP

26.6.3. World Wide Web

Résztvevők: kiszolgálók és kliensek (pl. böngészők). Formátum: HTML. Protokoll: HTTP.

Gyorsító technika: cache (kliensnél, lokális hálózatban vagy a szolgáltatónál) – timeout.

26.6.4. Peer-to-Peer

Egyenértékű partnerek közötti kapcsolat. Nincs központi irányítás, sem megbízható partner.

Pl. Napster (központi információs szerverrel), Gnutella (elosztott lekérdezések), BitTorrent, ...