

1. lecture

1. Mi a **hálózati host**?

Olyan eszköz, amely egy számítógépes hálózattal áll összeköttetésben. Információt oszthat meg, szolgáltatásokat és alkalmazásokat biztosíthat a hálózat további csomópontjainak.

2. Mi a **átviteli csatorna**?

Az a közeg, amelyen a kommunikáció folyik a résztvevő hosztok között. Ez a közeg lehet egy koaxális kábel, a levegő, optikai kábel, stb.

3. Definiálja a **propagációs késést**.

Az az időtartam, amely a jelnek szükséges ahhoz, hogy a küldőtől megérkezzen a címzetthez.

Jelölése: d_{prop} vagy d

4. Definiálja az **átviteli késleltetés**.

Az az időtartam, amely egy csomag összes bitjének az átviteli csatornára tételéhez szükséges.

Jelölése: d_T

5. Definiálja a **jel sávszélességet**.

Jel feldolgozás esetén az egymást követő frekvenciák legnagyobb és legkisebb eleme közötti különbséget nevezik jel sávszélességnek.

Tipikusan Hertz-ben mérik.

6. Definiálja a **hálózati sávszélességet**.

Az adat átviteléhez elérhető vagy felhasznált kommunikációs erőforrás mérésére szolgáló mennyiség, amelyet bit per másodpercben szoktak kifejezni.

7. Mi a fő különbség a **áramkörkapcsolt** és az **csomagkapcsolt** hálózatok között?

Áramkörkapcsolt: pl a telefon, egy hoszt dedikált erőforrást használ, az erőforrást le kell foglalni.

Csomagkapcsolt: csak rá kell tenni a csomagokat a hálózatra, és az állomások maguk döntenek el, merre továbbítják (nem kell lefoglalni az erőforrást, megosztott használat)

8. Sorolja fel a **hálózati kiterjedéseket**.

PAN: Personal Area Network (1 m²)

LAN: Local Area Network (10-1000 m²)

MAN: Metropolitan Area Network (10 km²)

WAN: Wide Area Network (100-1000 km², de az internet is)

9. Mit jelent a **legjobb szándék (best effort)** elv a hálózati kommunikációban?

Ha egy csomag nem éri el a célt, akkor törlődik, ilyenkor az alkalmazás újraküldi.

10. Mit jelent a **Black-box** megközelítés a kapcsolatokra?

Az eszközök (black box, később gateway, router) nem őrzik meg a csomaginformációkat, nincs folyam-felügyelet

11. Sorolja fel az **internet 5 (előadáson elhangzott) jellemzőjét.**

Rendszerfüggetlenség

Nincs központi felügyelet

LAN-okból áll

Globális

Szolgáltatásokat nyújt, pl WWW, e-mail, fájlátvitel

12. Hány réteget különböztet meg az **ISO/OSI referencia modell**? Sorolja fel őket.

#		
7.	Alkalmazási	Application
6.	Megjelenítési	Presentation
5.	Munkamenet/Ülés	Session
4.	Szállítói	Transport
3.	Hálózati	Network
2.	Adatkapcsolati	Data Link
1.	Fizikai	Physical

13. Hány réteget különböztet meg a **Tannenbaum**-féle hibrid rétegmodell? sorolja fel őket.

#		
5.	Alkalmazási	Application
4.	Szállítói	Transport
3.	Hálózati	Network
2.	Adatkapcsolati	Data Link
1.	Fizikai	Physical

sima TCP/IP modellben a fizikai és adatkapcsolati réteg egy kapcsolati réteggént jelenik meg

14. Mi az "Open System Interconnection Reference Model" (**ISO/OSI**), hogyan specifikáljuk az egyes rétegeket?

Open System Interconnection Reference Model: 7 rétegű standard, koncepcionális modellt ad meg kommunikációs hálózatok belső funkcionalitásához.

Réteg:

szolgáltatás (mit csinál)

interfész (hogyan férhetünk hozzá)

protokoll (hogyan implementáljuk)

15. Mi a feladata és mik a főbb funkcionálisai az ISO/OSI modell **fizikai rétegének**?

- Szolgáltatás
 - Információt visz át két fizikailag összekötött eszköz között
 - Definiálja az eszköz és a fizikai átviteli közeg kapcsolatát
- Interfész
 - Specifikálja egy bit átvitelét
- Protokoll
 - Egy bit kódolásának sémája
 - Feszültség szintek
 - Jelek időzítése
- Példák
 - koaxiális kábel
 - optikai kábel
 - rádió frekvenciás adó

16. Mi a feladata és mik a főbb funkcionálisai az ISO/OSI modell **adatkapcsolati rétegének**?

- Szolgáltatás
 - Adatok keretekre tördelése: határok a csomagok között
 - Közeghozzáférés vezérlés (MAC)
 - Per-hop megbízhatóság és folyamvezérlés
- Interfész
 - Keret küldése két közös médiumra kötött eszköz között
- Protokoll
 - Fizikai címezés (pl. MAC address, IB address)
- Példák
 - Ethernet
 - Wifi
 - InfiniBand

17. Mi a feladata és mik a főbb funkcionálisai az ISO/OSI modell **hálózati rétegének**?

- Szolgáltatás
 - Csomagtovábbítás
 - Útvonalválasztás
 - Csomag fragmentálás kezelése
 - Csomag ütemezés
 - Puffer kezelés
- Interfész
 - Csomag küldése egy adott végpontnak
- Protokoll
 - Globálisan egyedi címeket definiálása
 - Routing táblák karbantartása
- Példák
 - Internet Protocol (IPv4)
 - IPv6

18. Mi a feladata és mik a főbb funkcionálisai az ISO/OSI modell **ülés/munkamenet rétegének**?

- Szolgáltatás
 - kapcsolat menedzsment: felépítés, fenntartás és bontás
 - munkamenet típusának meghatározása
 - szinkronizációs pont menedzsment (checkpoint)
- Interfész
 - Attól függ...
- Protokoll
 - Token menedzsment
 - Szinkronizációs checkpoints beszúrás
- Példák
 - nincs

19. Mi a feladata és mik a főbb funkcionálisai az ISO/OSI modell **megjelenítési rétegének**?

- Szolgáltatás
 - Adatkonverzió különböző reprezentációk között
 - Pl. big endian to little endian
 - Pl. ASCII to Unicode
- Interfész
 - Attól függ...
- Protokoll
 - Adatformátumokat definiál
 - Transzformációs szabályokat alkalmaz
- Példák
 - nincs

20. Mi a feladata és mik a főbb funkcionálisai az ISO/OSI modell **alkalmazási rétegének**?

- Szolgáltatás
 - Bármilyen...
- Interfész
 - Bármilyen...
- Protokoll
 - Bármilyen...
- Példák
 - kapcsolj be a mobilod és nézd meg milyen appok vannak rajta...

2. lecture

21. Mit jelent a hálózatok esetén az **adatok burkolása**?

Mindegyik réteg hozzáteszi a saját fejlécét az üzenethez, amely réteg-specifikus infókat tartalmaz interfészek definiálják a rétegek közti interakciókat, a rétegek csak az alattuk lévőkre épülnek pl. a fizikai réteg nem tud az alkalmazásról, az alkalmazásnak nem kell törődnie a fizikaival

22. Adjon egy valós példát **adatok beburkolására** (pl. az előadáson látott Internet példa)!

Ethernet Header	IP Header	TCP Header	HTTP Header	Web Page	Ethernet Trailer
-----------------	-----------	------------	-------------	----------	------------------

23. Mit értünk **Internet homokóra** alatt? Miért nehéz az IPv6-ra való átállás?

Az Internet rétegnek hála, minden hálózat képes együttműködni. Minden alkalmazás működik minden hálózaton. Ezen réteg felett és alatt lehetnek újabb fejlesztések. Azonban az IP-t lecserélni nagyon nehéz pontosan az összekötő szerepe miatt.

24. A Hálózati réteg funkcióit milyen **síkok (planes)** mentén csoportosíthatjuk még?

Control plane (vezérlési sík): hogyan határozzuk meg az útvonalat?

Data plane (adat sík): hogyan továbbítjuk az adatot egy útvonal mentén?

25. Jellemezze egy mondatban a **tűzfalakat**, **proxykat** és **NAT dobozokat**!

Tűzfal: védelmi rendszer, az alkalmazási réteg fejléceit is vizsgálnia kell

Proxyk: alkalmazási végpontot szimulál a hálózatban

NAT doboz: megtöri a végpont-végpont elérhetőséget a hálózatban

26. Mi a **szimbólumráta** és az **adatráta**? Mi a mértékegységük?

Szimbólumráta: elküldött szimbólumok száma másodpercenként (Baud)

Adatráta: elküldött bitek száma másodpercenként (bps)

Egy szimbólum állhat több bitből

27. Mit mond ki a **Nyquist tétel**?

Zajmentes csatorna esetén a

Max adatsebesség = $2H * \log_2(V)$ bps

H: sávszél

V: szimbólumok száma

28. Mit mond ki a **Shannon tétel**?

Zajos csatornán

Max adatsebesség = $H * \log_2(1 + S/N)$ bps

H: sávszél

S/N: jel-zaj teljesítményének hányadosa

29. Ismertesse a fizikai rétegben a lehetséges **átviteli közegek** fajtáit!

Mágneses adathordozó (merevlemez)

Sodort érpár (távbeszélőrendszerek)

Koaxális kábel (nagyobb sebesség és távolság)

Fénykábel (fényforrás, közeg, detektor)

Rádiófrekvenciás (egyszerű, nagy táv, frekvenciafüggő terjedés)

Mikrohullámú (egyenes vonal mentén terjed, elhalkulás problémája, olcsó)

Infravörös átvitel (kis táv, szilárd tárgyakon nem hatol át)

Látható fény (lézer, nagy sávszél, olcsó, nem engedélyköteles, időjárásfüggő)

Műholdas

30. Mit nevezünk **frekvenciának**? Hogyan jelölik? Mi a mértékegysége?

Az elektromágneses hullám másodpercenkénti rezgésszáma.

Jelölése: **f**

Mértékegysége: Hertz (**Hz**)

31. Mi a **hullámhossz**?

Két egymást követő hullámcsúcs (vagy hullámvölgy) közti távolság

Jelölése: **λ**

32. Mi a **fénysebesség**?

Elektromágneses hullámok terjedési sebessége vákuumban

Jelölése: **c** = $3 \cdot 10^8$ m/s

részben és üvegszálaban ennek csak 2/3-a

33. Összefüggés **hullámhossz, frekvencia és fénysebesség** között

hullámhossz \times frekvencia = fénysebesség

34. Soroljon fel 3 **elektromágneses tartományt a frekvenciáik növekvő sorrendjében!**

1. Rádió
2. Mikrohullám
3. Infravörös
4. Látható
5. Ultraibolya
6. Röntgensugár
7. Gammasugár

35. Milyen frekvencia tartomány átvitelére alkalmas a **sodort érpár**, a **koax kábel** és az **optikai szál**?

Közeg	Frekvenciatartomány
Sodort érpár	104 - 106 Hz
Koax kábel	105 - 108 Hz
Optikai szál	1014 - 1015 Hz

Soroljon fel **3 óraszinkronizációs módszert!**

- **Explicit órajel**
Párhuzamos átviteli csatornák használata esetén explicit küldjük az órajeleket; rövid átvitele esetén alkalmas
- **Kritikus időpontok**
Adott időkor sync, pl szimbólum v blokk kezdetén ezen kívül az órák szabadon futnak, remélhetőleg szinkronban
- **Önütemező jel**
Külön órajel sync nélkül dekódolható jel, a szignál tartalmazza a szinkronizáláshoz szükséges infót

36. Ismertesse az **NRZ-L** (Non-Return to zero) kódolás szabályait!

Bit	Jel
1	Magas
0	Alacsony

Deszinkronizációra hajlamos

37. Ismertesse a **Manchester** kódolás szabályait!

Bit	Jel
1	Magasról alacsonyra
0	Alacsonyról magasra

Nincs óraelcsúszás, de az átvitel felét használja csak ki (két óraidő ciklus kell egy bithez)

Példa:

Bit	0	0	1	1	0
Man	—/—	—/—	—\—	—\—	—/—
Clock	—	—	—	—	—

38. Ismertesse az **NRZI** (Non-return to zero inverted)? Mi a fő probléma ezzel a kódolással?

Bit	Jel
1	Vált
0	Tartja a szintet

A csupa egyes sorozat problémáját megoldja ugyan, de a csupa nulla sorozatot ez sem kezeli

Példa:

Bit	0	0	1	0	1	0	1	1	0	0
NRZI	—	—	/	—	\	—	/	\	—	—
Clock	—	—	—	—	—	—	—	—	—	—

39. Ismertesse a **4-bit/5-bit módszert**. Miért van erre szükség? Hol használjuk?

Minden 4 bitet 5 bitbe kódolunk úgy, hogy elején max 1, végén max 2 nulla lehet elkerüli a csupa 0 sorozatokat, ahol az NRZI elcsúszhat.

Hátrányok: 20%-os hatékonyságvesztés

40. Mik a főbb tulajdonságai a **baseband** (alapsávú) átvitelnek?

a digitális jel direkt árammá vagy feszültséggé alakul

a jel minden frekvencián átvitelre kerül

átviteli korlátok

41. Ismertesse a digitális **alapsávú átvitel** struktúráját!

1. Forrás
2. Forrás kódolás (forrás bitek)
3. Csatorna kódolás (csatorna szimbólumok)
4. Fizikai átvitel
5. Médium
- vissza 4->3->2->1, minden lépés dekódolása

42. Mik a főbb tulajdonságai a **broadband** (szélessávú) átvitelnek?

Széles frekvenciatartományban történik az átvitel

Jelmodulációs lehetőségek:

- **Vivőhullámra ültetés** - amplitúdó moduláció
- **Vivőhullám megváltoztatása** - frekvencia vagy fázis moduláció
- **Különböző vivőhullámok felhasználása egyidejűleg**

43. Ismertesse a digitális **szélessávú átvitel** struktúráját!

1. Forrás
2. Forrás kódolás (forrás bitek)
3. Csatorna kódolás (csatorna szimbólumok)
4. Moduláció (Hullámformák véges halmaza)
5. Fizikai átvitel
6. Médium
- vissza 5->4->3->2->1, minden lépés dekódolása

44. Mi az **amplitúdó moduláció**?

A küldendő **s(t)** szignált a szinuszgörbe amplitúdójaként kódoljuk:

$$fA(t) = s(t) * \sin(2\pi * f * t + \varphi)$$

t: periódus idő

f: frekvencia

s(t): amplitúdó

φ: eltolás

Digitális jelnél a szignál erőssége egy diszkrét halmaz értékeinek megfelelően változik (pl.: 0-1)

45. Mi a **frekvencia moduláció**?

A küldendő **s(t)** szignált a szinuszgörbe frekvenciájaként kódoljuk:

$$fF(t) = a * \sin(2\pi * s(t) * t + \varphi)$$

t: periódus idő

s(t): frekvencia

a: amplitúdó

φ: eltolás

46. Mi a **fázis moduláció**?

Az **s(t)** szignált a szinuszgörbe fázisában kódoljuk:

$$fP(t) = a * \sin(2\pi * f * t + s(t))$$

t: periódus idő

f: frekvencia

a: amplitúdó

s(t): eltolás

3. lecture

47. Ismertesse a **médium többszörös használatának 5 módszerét!**

- **SDM - Térbeli** multiplexálás (**S**pace-**D**ivision **M**ultiplexing)
Külön vezeték vagy antenna
- **FDM - Frekvencia** multiplexálás (**F**requency-**D**ivision **M**ultiplexing)
Több szignál kombinációja adja az átvitelt, minden szignálhoz más frekvencia tartozik
- **WDM - Hullámhossz** multiplexálás (**W**avelength-**D**ivision **M**ultiplexing)
Optikai kábeleknél használt
- **TDM - Időbeli** multiplexálás (**T**ime-**D**ivision **M**ultiplexing)
Jelsorozat időintervallumokra szegmentálása, minden állomás saját időszeletet kap
- **CDMA - Kód** multiplexálás (**C**ode **D**ivision **M**ultiple **A**ccess)
Állomások egyfolytában sugározhatnak a teljes frekvenciasávon
Feltételezi, hogy a többszörös jelek lineárisan összeadódnak
Kulcs: a hasznos jel kiszűrése

48. Mi a **CDMA**? Ismertesse a működési algoritmusát.

A kódosztásos többszörös hozzáférés (angolul Code Division Multiple Access, röviden CDMA) a multiplexálás egy formája és a többszörös hozzáférés egy lehetséges megvalósítása, amely az adatokhoz csatornánként speciális kódokat rendel, és kihasználja a konstruktív interferencia tulajdonságot a multiplexáláshoz.

Algoritmus:

1. Minden bitidőt *m* darab intervallumra osztunk (chip)
2. Minden állomásnak van egy *m* bites kódja (chip sequence - páronként ortogonálisak)
3. **1**-es bitet akar továbbítani egy állomás: saját töredéksorozatát küldi tovább
4. **0**-ás bitet akar továbbítani egy állomás: saját töredéksorozatát 1-es komplementjét küldi tovább

49. Mi az a **Walsh mátrix**? Mire használható?

Oszlopai vagy sorai meghatároznak egy kölcsönösen ortogonális chipkód halmazt, CDMA multiplexálásra
Mérete csak 2 hatványai lehetnek, elemei +1 és -1 lehetnek.

50. Hogyan áll elő a **H(2k)**-nal jelölt Walsh mátrix?

$$H(2k) = [[H(2k-1), H(2k-1)], [H(2k-1), -H(2k-1)]]$$

$$\begin{bmatrix} H(2^{k-1}), & H(2^{k-1}) \\ H(2^{k-1}), & -H(2^{k-1}) \end{bmatrix}$$

Ahol

$$H(2^1) = [[1, 1], [1, -1]]$$

$$H(2^2) = [[[1, 1], [1, -1]], [[1, 1], [1, -1]], [[1, 1], [1, -1]], [[-1, -1], [-1, 1]]]$$

51. Melyek az **adatkapcsolati réteg** legfontosabb feladatai?

- Jól definiált szolgálati interfész a hálózati rétegnek
 - Nyugtázatlan összeköttetés alapú
 - Nyugtázott összeköttetés nélküli
 - Nyugtázott összeköttetés alapú
- Átviteli hibák kezelése
- Adatforgalom szabályozása (elárasztás elkerülése)

52. Milyen módszereket ismer a **keretezésre** az adatkapcsolati rétegben?

Bájt alapú: karakterszámlálás, bájtbeszúrás

Bit alapú: bitbeszúrás

Óra alapú: SONET - (Synchronous Optical Network) kódolás

53. Hogyan működik a **karakterszámlálás**?

Keret fejlécében megadjuk a keretben lévő karakterek számát

Érzékeny a hibára

54. Hogyan működik a **karakterbeszúrás** (bájt beszúrás)?

Keret elején végén *FLAG* byte, + *ESC* byte

A beszúrás előtt már szereplő *FLAG* és *ESC* bájtok elé *ESC* bájtokat teszünk

55. Hogyan működik a **bit beszúrás**?

Minden keret speciális bitmintával kezdődik és végződik: *01111110* (High-level Data Link Protocol (HDLC) esetén)

A küldő az adatban előforduló minden *11111* részsorozat után berak egy *0*-t, a fogadó majd eltávolítja

56. Hogyan működik az **óra alapú** keretezés (pl. SONET)?

STS-1 keretei fix méretűek, 9*90 bájt, a fogadása után újabb keret kezdő mintázatot kezd keresni

57. Mit tud mondani a bájt beszúrás és a **bit beszúrás hatékonyságáról** legrosszabb esetben?
- **Bájtbeszúrás**
50%, ha minden bájt flagbájt
 - **Bitbeszúrás** > Legrosszabb esetben 20% teljesítmény csökkenés, ha csak 1-esek
58. Mi az **egyszerű bithiba** definíciója?
- Az adat **1** bitje **1** helyett **0** lesz vagy fordítva
59. Definiálja a **csoportos bithibát** adott védelmi övezet (m) mellett!
- A fogadott bitek egy olyan folytonos sorozata, amelynek az első és utolsó bitje hibás, és nem létezik ezek közt olyan m hosszú részsorozat, amelyet helyesen fogadtunk volna (m = védelmi övezet)
60. Definiálja egy tetszőleges S kódkönyv **Hamming távolságát**?
- S kódkönyvben szerepeljenek egyenlő hosszú bitszavak, ekkor S Hamming-távolsága:
 $d(S) = \min \{ d(x,y) \mid x \neq y \text{ eleme } S \}$
61. Mi az a **Hamming korlát**?
- C**: kód (n hosszú szavakból)
 $|C| * \sum_{i=0}^{\lfloor (d(C)-1)/2 \rfloor} \binom{n}{i} \leq 2^n$
62. Mi a **kódráta** és a **kód távolság**? Milyen a rátája és távolsága egy jó kódkönyvnek?
- **Kódráta**
 $\log_2 |S| / n$
 Hatékonyságot adja meg
 - **Kódtávolság**
 $d(S) / n$
 Hibakezelési lehetőségeket adja meg
63. Milyen összefüggés ismeretes egy tetszőleges kódkönyv, a Hamming távolsága és **hibafelismerő** képessége között?
- d bithiba felismeréséhez a kódkönyv H-távolsága minimum $d+1$ legyen
64. Milyen összefüggés ismeretes egy tetszőleges kódkönyv, a Hamming távolsága és **hibajavítási** képessége között?
- d bithiba javításához a kódkönyv H-távolsága minimum $2d+1$ legyen
65. **Mikor érdemes hibajelző kódot és mikor hibajavító kódot** használni?
- Hibajelző**: megbízható hálózat (ARQ) - olcsóbb
Hibajavító: megbízhatatlan hálózat, gyakori hibákkal (FEC) - sok ismétlés elkerülésére

66. Hogyan működik a **Hamming kód** (több paritásos módszer)?

Egy paritást használó módszer

1. A kódszó biteit számozzuk meg 1-gyel kezdődően
2. 2 egészhatvány sorszámú pozíciói lesznek az ellenőrző bitek, azaz 1,2,4,8,16...
3. A maradék helyeket az üzenet biteivel töltjük fel
4. Mindegyik ellenőrző bit a bitek valamilyen csoportjának a paritását állítja be párosra (vagy páratlanra)
5. Egy bit számos paritásszámítási csoportba tartozhat:
 - k pozíciót írjuk fel kettő hatványok összegeként, a felbontásban szereplő ellenőrző pozíciók ellenőrzik a k -adik pozíciót
 - Példa: $k = 13$ -ra $k = 1 + 4 + 8$, azaz az első, a negyedik illetve a nyolcadik ellenőrző bit fogja ellenőrizni

67. Mi a **redundancia** szerepe a hibafelügyeletben?

Egy keret felépítése

- m adat bit
ez az üzenet
- r redundáns / ellenőrző bit
az üzenetből lett kiszámolva, új információt nem tartalmaz
- A keret teljes hossza: $n = m + r$
Az így előálló n bites bitsorozatot n hosszú kódszónak nevezzük!

4. lecture

68. Mi a **CRC**? Mire használható?

Cyclic Redundancy Check, hibajelző kód

Bitsorozatokat **Z₂** feletti polinomok reprezentációjának tekinti

69. Ismertesse a **CRC**-t használó algoritmus 4 lépését!

1. Legyen $G(x)$ foka r . r darab 0 hozzáfűzése $M(x)$ -hez, így az $x^r M(x)$ lesz
2. Az ehhez tartozó bitsorozatot elosztjuk a $G(x)$ sorozatával mod 2
3. $x^r M(x)$ -ből vonjuk ki a **maradékot**, ez lesz $T(x)$, az ellenőrző összeggel ellátott, továbbítandó keret.
4. A vevő $T(x) + E(x)$ -et kapja, ezt elosztja $G(x)$ -szel. ha a maradék, $R(x)$, nem 0, akkor hiba történt

70. Mikor **nem ismeri fel** a hibát a vevő oldal?

A $G(x)$ többszöröseinek megfelelő bithibákat nem ismeri fel

71. **CRC** esetén mit lehet mondani **hibajelző képességéről**, ha a generátor polinom $x+1$ többszöröse?

ezen esetben minden páratlan számú hiba felismerhető

72. Mutassa be röviden a **korlátozás nélküli szimplex protokollt**!

– **Környezet:**

Adó és vevő hálózati rétegei mindig készen állnak

Feldolgozási idő nincs

Végtelen puffer

A csatorna hibátlanul továbbít

– **Protokoll:**

Nincs sorszám, nyugta Küldő végtelen ciklusban folyamatosan küld Vevő a keret érkezésekor az adatrészt továbbítja a hálózati rétegnek

73. Mutassa be röviden a **szimplex megáll-és-vár protokollt**!

– **Környezet:**

Adó-vevő mindig kész

van feldolgozási idő

Végtelen puffer

Hibátlan csatorna

– **Protokoll:**

Küldő egyesével küld, és addig nem küld újat, amíg nem kap nyugtát

A vevő várakozik a keretre, ha megjött, adatrészt továbbküldi a hálózati rétegnek, végül nyugtáz

– **Következmény:**

fél-duplex csatorna kell (nyugta miatt)

74. Mutassa be röviden a **szimplex protokollt zajos csatorna** esetén!

– **Környezet:**

Adó-vevő mindig kész

Van feldolgozási idő

Végtelen puffer

A csatorna hibázhat

– **Protokoll:**

A vevő egyesével küld, amíg nem kap nyugtát a határidőn belül; ha ez lejár, újraküld

A vevő várakozik, ha megjön, akkor csekkolja az ellenőrző összeget; ha ok, küldi fel, ha nem, eldobja és nem nyugtáz

Ha a nyugta elveszik, duplikátum! Megoldás: alternáló bit protokoll (keretek sorszámozása)

75. Mit nevezünk **adási ablaknak**?

A küldhető sorozatszámok halmaza

76. Mit nevezünk **vételi ablaknak**?

A fogadható sorozatszámok halmaza

77. Mutassa be röviden a **csúszóablak protokollt**!

Egyszerre több keret is küldési állapotban lehet.

A fogadó n keretnyi puffert foglal, a küldőnek max ennyi keretet küldhet ki nyugtázatlanul.

A keret sorozatbeli pozíciója adja a címkéjét.

A fogadó nyugtája tartalmazza a következő várt keret sorszámát (kumulatív nyugta...) A hibás és a nem jó számú kereteket eldobja

A küldő nyilvántartja a küldhető sorozatszámokat (adási ablak) A fogadó a fogadható sorszámokat (vételi ablak)

Az adási ablak minden küldéssel szűkül, nyugtával nő

78. Mi a **visszalépés N-nel stratégia** lényege?

A hibás keret utáni kereteket a fogadó eldobja, és nem is nyugtázza. Az adó a timeout lejárta után újraküldi az összes nyugtázatlan keretet. (1 méretű ablakot tételez fel a fogadó részéről) - nagy sávszél pazarlás, ha sok a hiba

79. Mi a **szelektív ismétléses stratégia** lényege?

A hibás keretet a fogadó eldobja, de az utána érkező jókat pufferelem. A küldő a timeout után a legrégebbi nyugtázatlan keretet küldi újra. NAK javíthat a hatékonyságon, egynél nagyobb vételi ablak kell

80. Mely 3 dolgot biztosítja a **PPP protokoll**?

- Keretezési módszert egyértelmű határokkal
- Kapcsolatvezérlő protokollt a vonalak felélesztésére, tesztelésére, az opciók egyeztetésére, és a vonalak elengedésére.
- Olyan módot a háózati réteg opcióinak megbeszélésre, amely független az alkalmazott hálózati réteg protokolltól.

81. A **csatorna kiosztásra** mik a legelterjedtebb módszerek?

- Statikus (FDM, TDM)
- Dinamikus
- Verseny vagy ütközés alapú (ALOHA, CSMA, CSMA/CD)
- Verseny-mentes (bittérkép alapú, bináris visszaszámlálás)
- Korlátozott verseny (adaptív fabejárás)

82. Röviden mutassa be a **frekvenciaosztásos nyalábolás** módszerét!

- N db userhez a sávszélt N egyenlő méretű sávra osztja
- Fix számú usernél, nagy forgalomigénynél jó
- Löketszerű forgalom esetén problémás

83. Röviden mutassa be az **időosztásos nyalábolás** módszerét!

- N db userhez az időegységet N egyenlő méretű időrésre osztja
- Löketszerűnél nem jó

84. A csatorna modellben mit nevezünk **ütközésnek**?

Ha két keret egyidőben kerül átvitelre, akkor átlapolódnak, és értelmezhetetlenné válnak

85. Írja le a **folytonos és a diszkrét időmodell** lényegét!

– **Folytonos**

Mindegyik állomás tetszőleges időpontban megkezdheti a kész keretének sugárzását

– **Diszkrét**

Az időt diszkrét résekre osztjuk, sugárzás csak az időrések elején lehetséges. Egy időrés lehet üres, sikeres vagy ütközéses.

86. Mit jelent a **vivőjel érzékelési (Carrier Sensing)** képesség?

Az állomások meg tudják vizsgálni a közös csatorna állapotát küldés előtt, hogy foglalt-e vagy szabad. Ha foglalt, addig nem próbálják meg használni. Ha nem rendelkeznek ezzel a képességgel, akkor küldenek, ahogy megvan rá a lehetőségük.

5. lecture

87. Hogyan működik az egyszerű **ALOHA** protokoll?

Ha van küldendő adat, akkor a host elküldi.

88. Mit jelent a **keretidő az ALOHA** protokoll esetén?

Egy fix hosszúságú keret átviteléhez szükséges idő

Keretfeldolgozási idő + átviteli késés + propagációs késés (T_f)

89. Mennyi az **Aloha** protokoll esetén az **áteresztőképesség** (átvitel) a terhelés függvényében?

$$S = G * e^{-2G}$$

S: átjutott keretek

G: összesen elküldött keretek

90. Mennyi a **réselt Aloha** protokoll esetén az **áteresztőképesség** a terhelés függvényében?

$$S = G * e^{-G}$$

S: átjutott keretek

G: összesen elküldött keretek

91. Hogyan működik a **réselt ALOHA** protokoll?

A csatornát azonos időrésekre bontjuk, egy időrés = T_f . Átvitel csak az időrések határán lehetséges

Algoritmus:

Amikor egy keret küldésre kész, akkor kiküldi a következő időrés határon

92. Mit nevezünk **sebezhetőségi időnek**?

Az az időtartam, amely alatt ha másik keret is elküldésre kerül, akkor az aktuális keret sérül.

93. Hogyan működik az **1-perzisztens CSMA** protokoll?

Folytonos időmodell

Küldés előtt belehallgat

Ha foglalt, akkor vár, amíg fel nem szabadul.

Ha szabad, küld

Ütközéskor véletlen ideig vár, majd újrakezdi a procedúrát

94. Hogyan működik a **nem-perzisztens CSMA** protokoll?

Folytonos időmodell

Küldés előtt belehallgat

Ha foglalt, akkor véletlen ideig vár, majd újrakezd

Ha szabad, küld

Ütközéskor véletlen ideig vár, majd újrakezdi a procedúrát

95. Hogyan működik a **p-perzisztens CSMA** protokoll?

Diszkrét időmodell

Küldés előtt belehallgat

Ha foglalt, akkor a következő időrésig vár, majd újra

Ha szabad, akkor p valószínűséggel küld. Ha mégse küld, akkor a következő időrésben megint p -vel küld. Ez addig megy, amíg el nem küldi, vagy más nem kezd el küldeni. Ekkor úgy viselkedik, mintha ütközés történt volna.

Ütközéskor véletlen ideig vár, majd újra

96. Hogyan működik a **CSMA/CD protokoll**? (CD -> Collision Detection: ütközés érzékelés)

Egy CSMA protokoll kiegészítése így:

Minden állomás küldés közben is figyeli a csatornát, ha ütközést tapasztal azonnal megszakítja az adást (nem adja le a teljes keretet), véletlen ideig vár, majd újraküld. Újraküldés során a **binary exponential backoff** módszer alkalmazása

Nincs szükség nyugtára, mert az állomások észlelik az ütközést.

97. Mi a **Binary exponential backoff** módszer?

Válasszunk $[0, 2n-1]$ -ből egyet, ahol n az ütközések száma

Ennyi keretidőt várjunk az újraküldésig

n felső határa 10, 16 sikertelen próba után eldobjuk

98. Hogyan működik az **alapvető bittérkép eljárás**?

Versengési periódus N időrés, az i -edik hoszt ha küldeni akar, akkor az i -edik időrésben szór egy 1-et. A versengési periódus végére mindenki ismeri a küldőket, így sorban küldenek

99. Hogyan működik a **bináris visszaszámlálás protokoll**?

Minden állomásnak van azonos hosszú bitsorozat azonosítója, a versengési időben elkezdik bitenként küldeni az azonosítót, ha valaki 0-t küld de 1-et hall vissza a vagyolódás miatt, akkor lemond a küldési szándékáról

100. Mi a **korlátozott verseny protokollok** célja?

Ötvözni a versenyhelyzetes és a versenymentes protokollok jó tulajdonságait

Kis terhelés esetén versenyhelyzetes technikát használ a kis késleltetés érdekében, nagy terhelés esetén mellett ütközésmentes technika a csatorna jó kihasználása miatt

101. Hogyan működik az **adaptív fabejárás protokoll**?

Állomások bináris fában reprezentálva 0. időrésben mindenki küld

Ha ütközés, akkor mélységi bejárás, minden rés egy csomópont-hoz van rendelve

Ütközés esetén megnézzük a bal és a jobb csomópontot
Ha nincs ütközés, akkor a csomópont keresése befejeződik

102. Mi a **repeater** (ismétlő), és mire használják?

Analog eszköz, mely két kábelszegmenshez csatlakozik. Felerősíti a jelet és továbbítja. (fizikai réteg)

103. Mi az **hub** (elosztó) és mire használják?

Több bemenettel rendelkezik
A beérkező keretet minden vonalon továbbítja
Ha két keret egyszerre érkezik, ütközni fognak
Általában nem erősíti a jelet (fizikai réteg)
Olcsó, egyszerű, kevés tudással rendelkezik

104. Mi a **bridge** (híd), és mire használják?

Az adatkapcsolati rétegben működő eszköz, amely LAN-ok összekapcsolását végzi - lekorlátozzák az ütközési tartományok méretét
A bejövő keretet csak a megfelelő LAN-hoz továbbítja (forgalomirányítás az adatkapcsolati rétegben).
A portok külön ütközési tartományt képeznek és különböző sebességű hálózatokhoz csatlakozhatnak.
Pufferelést, csomagfeldolgozást végez, továbbító táblázatot (forwarding table) tart karban. Képest megtanulni a csatlakozó eszközök címét.

105. Mi a **backward learning** (Címek tanulása) lényege?

A hidak használják ezt a módszert a keretek továbbításához használt táblázatuk feltöltésére.
Ha egy keret érkezik hozzájuk, megnézik a forráscímet (feladót) és "megtanulják", hogy az melyik porton érhető el (ahonnan a keret jött), és ezt bejegyzik a táblázatukba.

106. Ismertesse a **feszítőfa protokoll** (STP) lépéseit?

1. Az egyik bridge a gyökér
2. Minden bridge megkeresi a legrövidebb utat hozzá
3. Ezen utak uniója a feszítőfa

A faépítés során a bridgek BPDU-kat (Configuration Bridge Protocol Data Unit-okat) cserélnek
Bridge ID, Gyökér ID, költség rendelődik a gyökérhez

A fogadása után a bridge választ egy új gyökeret, megjegyzi a felé vezető portot és a következő bridge-t felé

6. lecture

107. Mi a **forgalomirányító algoritmusok** definíciója?

A hálózati réteg szoftverének azon része, amely eldönti, hogy a bejövő csomag melyik kimeneti vonalon kerüljön továbbításra.
táblázatok feltöltése, karbantartása + irányítás

108. Mi a **statikus (nem adaptív) forgalomirányító algoritmusok** fő jellemzője?

Offline meghatározza előre a döntéseket, a router indulásakor - nem befolyásolja a topológia és a forgalom változása

109. Mi az **adaptív forgalomirányító algoritmusok** fő jellemzője?

A topológia és a forgalom is befolyásolhatja a döntést

110. Mit mond ki az **optimalitási elv** (forgalomirányítás esetén)?

Ha J az I->K optimális útvonalon van, akkor J->K optimális útvonal is ugyanerre esik.

Következmény: az összes forrásból egy célba tartó optimális utak egy nyelőfát alkotnak, aminek a gyökere a cél.

111. Mi a **távolságvektor (distance vector) alapú forgalomirányítás** lényege?

A routerek karbantartanak egy táblázatot, amiben minden célhoz szerepel a legrövidebb ismert távolság, és annak a vonalnak az azonosítója, amelyiken a célhoz el lehet jutni. Ezt a táblát a szomszédoktól kapott infók alapján frissítik (a routerek periodikusan elküldik a szomszédoknak a távolságvektorukat), amikor nem változik semmi már, az algónak vége.

112. Mi az **elosztott Bellman-Ford algoritmus**?

A távolságvektorok tábláinak frissítésére alkalmas

Minden csomópont csak a közvetlen szomszédjaival kommunikálhat

Aszinkron

Minden csomópont ismeri a közvetlen szomszédjaihoz a költségét, ezeket küldi tovább. Ez alapján aktualizálja minden csomópont a saját tábláját

113. Magyarázza el a **végtelenig számolás** problémáját!

Ha egy állomás (A) meghibásodik a közvetlen szomszédja (B) észleli, hogy a költség végtelen lett, mert nem érkezik A-tól csomag. B-nek egy szomszédja (C), amelyik korábban B-n keresztül érte el A-t, elküldi A elérési költségét. B azt fogja hinni, hogy C-n keresztül A elérhető, és a C-től kapott költséget megnöveli B-C költséggel, majd ezt küldi vissza C-nek.

Ezután mindketten folyamatosan azt fogják hinni, hogy a másikon keresztül A elérhető, és mindenlépésben B-C költséggel növelik A elérési költségét a táblázatukban.

114. Mik a **link-state (kapcsolatállapot) alapú forgalomirányítás** megvalósításának lépései?

1. szomszédok címének felkutatása: HELLO csomag szórása, a szomszédok válaszolnak a saját címükkel
2. késleltetés meghatározása: ECHO csomag küldése, a másik oldalnak azonnal vissza kell küldenie - körbeérési idő fele kb a késleltetés
3. infócsomag összeállítása: feladó azon., sorszám, korérték és a szomszédok listája a késleltetésekkel.
4. szétküldés elárasztással. a routerek megjegyeznek minden (sorszám,forrás) párt, és csak akkor küldik tovább, ha új
5. Dijkstra algoritmus lefuttatása ha minden infó megérkezett

115. Hasonlítsa össze a **távolságvektor alapú** és a **link-state (kapcsolatállapot) alapú** forgalomirányítást.

Az első esetében a routerek minden más routerre vonatkozó általuk ismert költséget elküldenek, de csak a közvetlen szomszédoknak, a második esetében csak a szomszédokra vonatkozó ismert költségeket küldik el mindenkinek.

116. Mi a **hierarchikus forgalomirányítás** lényege?

Nagy hálózaton a forgalomirányító táblák arányosan nőnek ezért alkalmazzunk hierarchikus forgalomirányítást:

A routereket tartományokra osztjuk. Minden router ismeri a sajátját, de a többi belső szerkezetéről nem tud

Többszintű hierarchia is lehetséges:

N routerből álló alhálózathoz optimálisan $\ln(N)$ szint kell, amely routerenként $e \cdot \ln(N)$ bejegyzést igényel

117. Mit nevezünk **broadcasting**nak (adatszórásnak)?

Egy csomag mindenhová történő egyidejű elküldése

118. Sorolja fel az **adatszórás megvalósítási lehetőségeit**.

- **Külön csomag küldése minden egyes címzettnek**
Sávszélt pazarol, lista kell
- **Elárasztás**
Kétpontos kommunikációhoz nem megfelelő
- **többcélú forgalomirányítás (multidestination routing)**
csomagban van egy lista a rendeltetési helyekről, a router a kimenő vonalakhoz készít egy másolatot, a másolatokba csak a megfelelő célcím listát írja be
- **Forrás routerhez tartozó nyelőfa használata**
Ha minden router ismeri, hogy mely vonalai tartoznak a fészékhez, akkor csak azokon továbbítja az adatszóró csomagot (kivéve amelyen érkezett)
- **Visszairányú továbbítás (reverse path forwarding)**
A router ellenőrzi, hogy azon a vonalon kapta-e meg a csomagot, amelyen rendszerint ő szokott az adatszórás forrásához küldeni. ha igen, akkor valószínűleg a csomag a legjobb utat követte idáig a forrástól, így ez az első csomag, ami megjött, szóval kimásolja minden vonalra.

119. Mit nevezünk **multicasting**nak (többsküldésnek)?

Egy csomag meghatározott csoporthoz történő egyidejű elküldése

Csoportkezelés is kell hozzá:

létrehozás, megszüntetés, csatlakozás, leválasztás a router a bejövő csomagot csak a fészék azon élein küldi tovább, amelyek csoporton belüli hosthoz vezetnek

120. Mire szolgál a **DF bit** az IPv4 fejlécében?

Ne darabold, *don't fragment flag* a routernek

A beérkező datagramot ne darabolja fel

121. Mire szolgál a **MF bit** az IPv4 fejlécében?

More fragment, jelzi, hogy még az aktuális datagramhoz ez nem az utolsó darab, azaz van még több is. (sorszám)

122. Mire szolgál az **azonosító** (azonosítás) az IPv4 fejlécében?

Datagram azonosítására szolgál, egy datagram összes darabja ugyanazt az azonosítót hordozza

123. Mire szolgál a **darabtolás** (fragment offset) az IPv4 fejlécében?

A darab helyét mutatja meg a datagramon belül

124. Mire szolgál az **élettartam** (TTL) mező az IPv4 fejlécében?

Time To Live, minden ugrásnál eggyel csökkenti a router az értékét, ha eléri a nullát, a csomagot eldobja

125. Mi az **IPv4** cím és hogyan ábrázoljuk?

Minden host és router az interneten rendelkezik egy IP címmel, amely a hálózat számát és a hoszt számát kódolja. ez a cím globálisan egyedi 4 bájtban ábrázoljuk, leírni bájtanként decimálisan ábrázolva, a bájtokat pontokkal elválasztva szoktuk

126. Milyen **IP cím osztályokat** ismer? Jelemezze ezeket!

- A: 0, hálózat(1), hoszt(3)
- B: 10, hálózat(2), hoszt(2)
- C: 110, hálózat(3), hoszt(1)
- D: 1110, multicast address
- E: 1111, jövőbeni felhasználásra

127. Milyen **speciális IPv4 címek** léteznek?

- 0...0
Az aktuális host
- 0...0, hoszt
Aktuális hálózaton lévő host
- 1...1
Broadcast a helyi hálózaton
- hálózat, 1...1
Broadcast távoli hálózaton
- 0111111, bármi (127. ...)
Visszacsatolás

128. Mi az **alhálózati maszk** és mire szolgál?

Segítségével elkülöníthető a hálózati azonosító és az állomás azonosító. Az IP cím hálózati részével megegyező hosszúsággal 1-est, utána 0-kat tartalmaz.

7. lecture

129. Mi az a **NAT doboz** és mire szolgál?

(Network Address Translation - hálózati címfordítás) Hálózati címfordítást végez. A vállalatban belüli globálisan nem egyedi IP címeket globálisan egyedi IP címre fordítja.

130. Mi az az **MTU** és mire szolgál?

MTU: Maximum Transmission Unit – lényegében a maximális használható csomag méret egy hálózatban.

131. Hogyan működik az **MTU felderítés**?

1. Csomagküldés a "don't fragment" flag bittel
2. Folyamatosan csökkentjük a csomag méretét, amíg egy meg nem érkezik

132. Hogyan ÉS hol történik az **fragmentált/darabolt IP csomagok** helyreállítása?

A végponton történik a hossz és eltolás (offset) ismeretében.

133. Mi az **IPv6** cím és hogyan ábrázoljuk?

Az IPv4 kibővítése. 128 bites címek, 4.8*1028 cím / ember. Formátum: 16 bites értékek 8 csoportba sorolva (':'-tal elválasztva), minden csoport elején szereplő nulla sorozatok elhagyhatók és csupa nulla csoportok elhagyhatók, ekkor '::'.

134. Mi a **localhost IPv6** esetén?

::1

135. Soroljon fel két olyan lehetőséget (az EA-on látott 4-ből), melyet az **IPv6 támogat, de az IPv4 esetén nem** találkoztunk vele?

- Forrás routing
- Mobil IP
- Privacy kiterjesztések
- Jumbograms

136. Mi gátolja az **IPv6-ra való átállást**?

IPv6 bevezetése a teljes Internet frissítését jelentené, minden router, minden host.

137. Hogyan oldható meg az **IPv6 csomagok átvitele IPv4 hálózat felett**?

Tunneleket használunk az IPv6 csomagok becsomagolására és IPv4 hálózaton való továbbítására.

138. Mire szolgál az **ICMP** protokoll?

Internet Control Message Protocol, váratlan események jelentésére használjuk. Többféle ICMP-üzenetet definiáltak:

- Elérhetetlen cél
- Időtúllépés
- Paraméter probléma
- Forráslefojtás
- Visszhang kérés
- Visszhang válasz

139. Mi lehet a hatása egy **ICMP forráslefojtás** csomagnak?

TODO

140. Mire szolgál az **ARP** és hogyan működik?

Feladata az IP cím megfeleltetése egy fizikai címnek. (Address Resolution Protocol) Adatszóró csomag kiküldése az Ethernetre "Ki-e a 192.60.34.12-es IP-cím?" kérdéssel az alhálózaton, és minden egyes host ellenőrzi, hogy övé-e a kérdéses IP-cím. Ha egyezik az IP a host saját IP-jével, akkor a saját Ethernet címével válaszol.

141. Mire szolgál a **RARP** és hogyan működik?

Feladata a fizikai cím megfeleltetése egy IP címnek. (Reverse Address Resolution Protocol) Az újonnan indított állomás adatszórással csomagot küld ki az Ethernetre "A 48-bites Ethernet-címem 14.04.05.18.01.25. Tudja valaki az IP címemet?" kérdéssel az alhálózaton. Az RARP-szerver pedig válaszol a megfelelő IP címmel, mikor meglátja a kérést.

142. Mi az a **DHCP** és hogyan működik?

Lehetővé teszi a dinamikus IP címkiosztást. A kliensek a DHCP esetén egy (megújítható) időszakra kapják az IP címet. A kiszolgáló másik LAN-on is lehet; DHCP közvetítő van LAN-onként

143. Milyen lehetőségeket támogat a **DHCP**?

IP címek kiosztása MAC cím alapján vagy dinamikusan. További hálózati paraméterek kiosztása (hálózati maszk, névkiszolgáló, domain név stb.)

144. Mi DHCP esetén a **cím bérlet**s?

A DHCP szerver a klienseknek az IP-címeket bizonyos bérleti időtartamra adja "bérbe".

145. Mi az **AS** (Autonóm rendszer)?

Az interneten autonóm rendszer (Autonomous System, AS), önálló rendszer vagy útválasztási tartomány (routing domain) IP-hálózatok (IP routing-prefixek) olyan csoportja, melyen belül egyetlen, jól meghatározott útválasztási irányelv (routing policy) érvényesül.

Egy adminisztratív tartomány alatti hálózat. Pl.: ELTE, Comcast, AT&T, Verizon, Sprint, ...

146. Miért van szükségünk **AS**-ekre?

- A routing algoritmusok nem elég hatékonyak ahhoz, hogy a teljes Internet topológián működjenek
- Különböző szervezetek más-más politika mentén akarnak forgalom irányítást (policy)
- Lehetőség, hogy a szervezetek elrejtse a belső hálózatuk szerkezetét
- Lehetőség, hogy a szervezetek eldöntsék, hogy mely más szervezeteken keresztül forgalmazzanak
- Röviden: egyszerűbb az útvonalak számítása, nagyobb rugalmasság és nagyobb autonómia/függetlenség.

147. Mi azonosít egy **AS**-t?

Minden AS-t egy AS szám (ASN) azonosít, ami egy 16 bites érték (a legújabb protokollok már 32 bites azonosítókat is támogatnak).

148. Milyen routing megoldást/protokollt alkalmaz a **BGP**?

Path vector (útvonalvektor) protokollt.

149. Hogyan működik az **útvonalvektor** protokoll?

A távolságvektor protokoll kiterjesztése. A teljes útvonalat meghirdeti (nem csak a következő ugrást).

150. Mit értünk az alatt, hogy minden AS **saját útválasztási politikát** alkalmazhat?

Az AS-en belüli protokoll független a többi AS-tól.

151. Sorolja fel az **IGP**, **iBGP** és **eBGP** szerepét?

- **IGP**
Útválasztás egy AS-en belül belső célállomáshoz
- **iBGP**
Útválasztás egy AS-en belül egy külső célállomáshoz
- **eBGP**
Routing információk cseréje autonóm rendszerek között

152. Mikor mondjuk **két AS-ről, hogy azok össze vannak kötve**?

Ha van közöttük a BGP routereiket összekötő vonal.

153. Adjon meg 3 példát **forgalomirányítási korlátozásra** AS-ek közötti routing esetén.

- Kereskedelmi forgalom ne menjen keresztül oktatási hálózaton.
- IBM forgalma ne menjen át a Microsoft-on.
- Albánián csak végszükség esetén haladjunk át.

154. Mit nevez a **BGP csonka hálózatnak**?

Olyan hálózatok, amelyeknek csak egyetlen összeköttetésük van a BGP gráffal.

155. Mit nevez a **BGP többszörösen bekötött hálózatnak**?

Olyan hálózatok, amelyeket használhatna az átmenő forgalom, de ezek ezt megtagadják.

156. Mit nevez a **BGP tranzit hálózatnak**?

Olyan hálózatok, amelyek némi megkötéssel, illetve általában fizetség ellenében, készek kezelni harmadik fél csomagjait.

157. Mire szolgál és hogyan működik a **VPN** (virtuális magánhálózat)?

TODO

8. lecture

158. Mire szolgál a **TCP** protokoll? Mik a főbb jellemzői?

Megbízható, sorrend helyes, kétirányú bájtfolyamok létrehozására (Transmission Control Protocol). Portszámok teszik lehetővé a demultiplexálást, kapcsolat alapú, folyamvezérlés, torlódásvezérlés, fair viselkedés. 20 bájtos fejléc + options fejlécek.

159. Mire szolgál az **UDP** protokoll? Mik a főbb jellemzői?

8 bájtos UDP fejléc (User Datagram Protocol). Egyszerű, kapcsolat nélküli átvitel. Portszámok teszik lehetővé a demultiplexálást, 16 bit → 65535 lehetséges port, 0 port nem engedélyezett. Kontrollösszeg hiba detektáláshoz, felismeri a hibás csomagokat viszont nem ismeri fel az elveszett, duplikátum és helytelen sorrendben beérkező csomagokat.

160. Hogyan történik egy **TCP kapcsolat** felépítése? Mik a lépései?

TCP flagek segítségével (SYN, ACK, FIN)

H1 → SYN → H2

H1 ← SYN ← H2

H1 ← ACK ← H2

H1 → ACK → H2

161. Hogyan történik egy **TCP kapcsolat lezárása**? Hogyan bomlik le?

A küldő jelzi a kapcsolat befejezését egy FIN szegmensben, és vár a szegmens nyugtájára. Az ellenkező irányban továbbra is lehet küldeni. Két félig lezárás zárja le a kapcsolatot.

162. Mit mondhatunk a **TCP átviteléről** az ablak és az **RTT** függvényében?

Az átvitel arányos a *ablak* / *RTT*-vel. Nagy ablakméret, vagy kis RTT esetén gyorsul az átvitel.

163. Mit jelent az **RTO**, és hol használják?

Ez szabályozza az időközt a küldés és egy duplikátum újraküldése között, ha egy nyugta kimarad. (Retransmission Timeout)

164. Hogyan történik az **RTT becslés** az eredeti TCP esetén?

RTT becslése mozgóátlaggal

$\text{new rtt} = \alpha * (\text{old rtt}) + (1 - \alpha) * (\text{new sample})$

Javasolt α : 0.8–0.9 (0.875 a legtöbb TCP esetén).

165. Mit mondhatunk **TCP esetén a hibadetektálásról**?

Hibamentes átvitelt biztosít. Mivel a csomaghibát, sorrendhibát, elveszett és a duplikált csomagot is detektálja.

166. Mi a fogadó által felajánlott **ablakméret (wnd)**?

A fogadó ezzel a mérettel jelzi, hogy mennyi adatot tud pufferelni, vagyis ez a csúszóablak protokollnál megismert fogadási ablak mérete. A tényleges ablakméretet (wnd) több paraméter határozza meg.

167. Mit jelent, ha a **fogadó wnd=0-át küld**?

A fogadó puffere tele van, átmenetileg nem tud több adatot fogadni (gyors adó).

168. Mit nevezünk **folyamvezérlésnek**?

A folyamvezérlés azt szabályozza, hogy a küldő milyen ütemezéssel küldheti az adatokat. Túl sok csomag túlterhelné a fogadót. A megoldás: csúszóablak.

169. Mit nevezünk **torlódásnak TCP esetén**?

Ha a terhelés túl nagy, túlszordulnak a pufferek, csomagok vesznek el, újra kell küldeni, drasztikusan nő a válaszidő. Ezt a torlódásnak nevezzük.

Amikor a hálózat terhelése nagyobb mint a kapacitás

170. Mi a **TCP Nagle** algoritmus működési alapelve?

1. Ha az ablak \geq MSS (Maximum Segment Size) és az elérhető adat \geq MSS, akkor küldjük el az adatot (egy teljes csomag küldése)
2. Különben, ha van nem nyugtázott adat, akkor várakoztassuk az adatot egy pufferben, amíg nyugtát nem kapunk
3. Különben, küldjük az adatot (küldjünk egy nem teljes csomagot, ha nincs más)

171. Mi a **TCP Karn** algoritmus? A kapcsolódó problémát is ismertesse!

Hogyan becsüljük meg az RTT-t? Megmérjük a küldés és a válasz között eltelt időt. A probléma: a válasz félreértelmezhető újraküldés esetén. Karn algoritmus: dobjuk el azokat a mintákat, melyek egy csomag újraküldéséből származnak.

172. Vázzolja a **TCP Incast problémát**!

Sok szimultán küldő egy fogadóhoz, ami miatt a switchek pufferei megtelítődnek és csomagok vesznek el és a nyugta nem megy vissza.

9. lecture

173. Mi az a **torlódási ablak**? Mire szolgál?

Congestion window, cwnd slow startnál használt második ablak. kezdetben MSS méretű

174. Mi az a **slow start** TCP esetén?

Exponenciális növekedés (hisztórikus elnevezés: korábban még agresszívebb sémák)

175. Mi az **AIMD TCP Tahoe** esetén?

Additive Increase Multiplicative Decrease (additív növelés, multiplikatív csökkentés). Az elküldhető csomagok számát (valójában cwnd-t) additív módon növeljük ha még nem értük el a hálózat kapacitását, és multiplikatív módon csökkentjük, ha már elértük. Ezt a stratégiát a slow start után, a torlódás elkerülési fázisban alkalmazza a TCP.

176. Mi a gyors újraküldéss **TCP RENO** esetén?

Ha csak egy csomag veszik el, akkor NEM várjuk meg a timeoutot (RTO), hanem újraküldjük a csomagot és folytatjuk a küldést. Az egy csomag elveszését a háromszoros nyugtaduplikátum jelzi. Ezt még a TCP Tahoe is ugyanígy csinálja.

177. Mit jelenthet az ha **három nyugta-duplikátum érkezik egymás után?**

Hogy valószínűleg elveszett egy csomag (lehet, hogy csak késik), de az utána következők megérkeztek, hiszen emiatt jönnek ugyanolyan sorszámú nyugták duplikátumai. Ha az utána következők is elvesztek volna, akkor semmilyen nyugta nem érkezne, hanem timeout lenne.

178. Mi a gyors visszaállítás **TCP Reno** esetén?

A gyors újraküldés után felezzük cwnd-t (ez a Multiplikatív Decrease), azaz nem állítjuk vissza 1-re (elkerüljük a lassú indulást). A TCP Tahoe ilyen esetben a lassú indulás fázisba lépne. Ha az RTO lejár akkor cwnd = 1 lesz (lassú indulás) Tahoe és Reno esetén is.

179. Mivel több a **TCP NewReno**? Mi a problémája az alkalmazott megoldásnak?

Minden duplikált ACK egy újabb csomag elküldését (nem újraküldést) váltja ki. Probléma: ha egy csomag > 3-mal eltér a sorrendjétől, az felesleges gyors helyreállítást és ezzel felesleges újraküldést okoz.

180. Mi a probléma **nagy késleltetés-sávszélesség szorzatú hálózatok** esetén?

Sok a szállítás alatt lévő adatmennyiség. A slow start és additív increase lassan konvergál.

181. Mely **TCP variánsok** használatosak napjainkban?

- TCP NewReno
- TCP Vegas
- Compound TCP
- TCP BIC
- TCP CUBIC

182. Hogyan működik a **Compound TCP**?

Reno alapú, két torlódási ablak, egy késleltetés alapú (dwnd) és egy vesztes alapú (cwnd). Ha nő az RTT, dwnd csökken, ha csökken az RTT, dwnd nő. $wnd = \min\{cwnd + dwnd, adv\ wnd\}$ Hátrány: folyamatos RTT becslést igényel.

183. Hogyan működik a **CUBIC TCP**?

Alap TCP implementáció Linux rendszereken. Az AIMD helyettesítése egy "köbös" (CUBIC) függvénnyel:

$$W_{cubic} = C * (T - K)^3 + W_{max}$$

C konstans

$$K = (W_{max} \beta / C)^{1/3}$$

184. Mik a TCP problémái kis folyamatok esetén?

Nincs lehetőség felgyorsulni a kevés adat miatt. Szinte végig a slow start fázisban marad.

185. Mik a **TCP problémái vezeték nélküli hálózatok esetén?**

Itt a csomagvesztés gyakoribb, és nem jelent torlódást, ahogy a TCP hiszi.

186. Mi a **DoS** támadás? Miért probléma ez TCP esetén?

Denial of Service. A TCP kapcsolatok állapottal rendelkeznek, a SYN csomagok erőforrást foglalnak le a szerveren. A DoS támadás abból áll, hogy meglehetősen sok SYN csomagot küldenek a szervernek, aminek a hatására elfogy a memóriája és összeomlik.

10. lecture

187. Mit nevezünk **munkamenet**nek az ISO/OSI referencia modellben?

Egy munkamenet a egymással összefüggő hálózati interakciók sorozata egy alkalmazási feladat elvégzése során.

188. Mik azok a **TLD**-k? Adjon meg 4 példát.

Top Level Domain

.com
.edu
.net
.org

189. A névfeloldásnál mit neveznek **iteratív lekérdezés**nek? Mik a jellemzői?

Ha a névszerver adja vissza a választ vagy legalább azt, hogy kitől kapható meg a következő válasz.

190. A névfeloldásnál mit neveznek **rekurzív lekérdezés**nek? Mik a jellemzői?

Ha a névszerver végzi el a névfeloldást, és tér vissza a válasszal.

191. Mit nevezünk **statikus weboldal**nak?

A statikus weboldal tartalma nem változik csak manuális átszerkesztéssel.

192. Mit nevezünk **dinamikus weboldal**nak?

A dinamikus weboldal valamilyen kód végrehajtásaként keletkezik, mint például: javascript, PHP, vagy mindkettő egyszerre.