

Vizsga felépítés

2024. december 18., szerda 9:56

- 60-70 perc
- **3 rész:**
 - **Angol szakkifejezések** - 20 kérdés, **50%!**
 - 10 perces, 1 p/k
 - **Teszt, rövid kérdések** - 20 kérdés, **55%!**
 - 20 perces, 1 p/k
 - **Esszé, hosszú kérdések** - 5 kérdés
 - 40 perces, 5 p/k (45p)

Értékelés:

85	100	5
75	85	4
60	75	3
50	60	2

1. előadás

2024. december 18., szerda 10:03

Hálózati sávszélesség:

- Adatátvitelhez használt kommunikációs erőforrás
- bit/sec (1 bájt = 8 bit)
- Szabványok:
 - SI mértékegységek: 10^{3n} ($n \geq 0$) bájtok (kilo, mega, giga, stb)
 - IEC: 2^{10n} ($n \geq 0$) bájtok (kibi, mebi, gibi stb)

Hálózati hoszt:

- Számítógépes hálózattal áll összeköttetésben
- Szolgáltatást, alkalmazásokat biztosít a további csomópontoknak

Átviteli közeg/csatorna:

- A közeg, ahol a kommunikáció folyik
- Koax/optikai kábel, levegő stb

Internet:

- WAN (Wide Area Network), globális
- Nincs központi felügyelet, LAN-okból (Local Area Network) áll
- Követelmények:
 - Hibatolerancia: több lehetséges útvonal a forrás és cél között
 - Rugalmasság: erőforrásmegosztás hatékonyság érdekében (linkek száma nem túl nagy)
 - Megfelelő csomópont kapacitás: linkek száma nem túl kicsi
- Topológia (kialakítás): switchelt hálózatok
 - Pros: Erőforrás megosztás és csomópontok kapacitása személyre szabható
 - Cons: Okos eszközöket igényel (forgalomirányítás, csomagtovábbítás)
- Tier-ek:
 - Tier 1: nemzetközi, nincs szolgáltató
 - Tier-2: nemzeti, 3-as szolgáltatóknak átjárás, legalább egy szolgáltató
 - Tier-3: helyi, legalább egy szolgáltató
- Peering:
 - Hálózatok közötti kölcsönös kapcsolat
 - Túl költséges lenne mindent mindennel
 - IXP (Exchange Points):
 - Több hálózat összekapcsolása egy fizikai helyen

Erőforráskezelés:

- Csúcsráta: P
- Átlagos ráta: A
- Előre foglalás
 - Előre lefoglaljuk a kellő sávszélességet (folyam szintű multiplexálás)
 - P-t foglalunk
 - A/P = kihasználtsági szint
- Igény szerinti
 - Akkor küldünk adatot, ha szükséges (csomag szintű multiplexálás)
 - Általában nagyobb kihasználtság
- Melyik jobb?
 - P/A kicsi => előre foglalás
 - P/A nagy => előre foglalni pazarló
- Resource Reservation Protocol:
 - Foglálási kérés
 - Áramkör kialakítása
 - Adatküldés
 - Lebontás

Pros	Cons
<ul style="list-style-type: none">• Kiszámítható teljesítmény• Egyszerű, gyors (miután kiépült)	<ul style="list-style-type: none">• Alacsony hatékonyság• Bonyolult felépítés/bontás• Hiba esetén új áramkör

- Csomagkapcsolás, csomagkapcsolt hálózatok:
 - Adatküldés csomagokban
 - Minden csomagban van célcím
 - Pufferelés szükséges

Pros	Cons
<ul style="list-style-type: none">• Hatékony• Egyszerű• Hibatolerancia	<ul style="list-style-type: none">• Kiszámíthatatlan telj.• Pufferelés kell

2. előadás

2024. december 18., szerda 10:51

Liskov principle:

- **Modularitás** az internet alapja
- Protokollokat rétegekbe rendezzük

Internet rétegmodelljei:

- **TCP/IP:** eleinte 4 réteg, később a hibrid változatban 5

Réteg	Protokollok
Alkalmazási (<i>Application</i>)	TELNET, FTP, HTTP, DNS
Szállítói (<i>Transport</i>)	TCP, UDP
Hálózati (<i>Internet</i>)	IP
Kapcsolati (<i>Link</i>)	ARPANET, Ethernet, LAN

- **OSI (referencia) model**
 - Open System Interconnection Reference Model
 - 7 réteg

Rétegek
Alkalmazási (<i>Application</i>)
Megjelenítési (<i>Presentation</i>)
Ülés (<i>Session</i>)
Szállítói (<i>Transport</i>)
Hálózati (<i>Network</i>)
Adatkapcsolati (<i>Data Link</i>)
Fizikai (<i>Physical</i>)

Rétegek saját adattípusa (PDU)

- **Fizikai:** *bit*
- **Kapcsolati:** *keretek*
- **Hálózati:** *csomagok*
- **Szállítói:** *szegmensek*
- **Alkalmazási:** *üzenet*
- Minden PDU-ban benne van az alatta lévő PDU tartalma

Hálózati kapcsolatok jellemzői:

- Késleltetés (csomagméret / kapcsolati sebesség)
- Adatvesztés
- Küldési sebesség

TCP/IP rétegek:

- **Alkalmazási:**
 - Szolgáltatások nyújtása
- **Szállítói:**
 - TCP vagy UDP protokollal kézbesítés
 - **TCP:**
 - Megbízható, kétirányú
 - Szegmentálás, felügyelet
 - **UDP:**
 - Nem megbízható, nincs felügyelet
 - Gyorsabb
- **Internet:**
 - Csomagtovábbítás, speciális csomagformátum
- **Kapcsolati réteg:**
 - Nem specifikált, LAN-tól függ

OSI rétegek:

- **Alkalmazási:**
 - Konkrétan akármilyen
 - Az első alkalmazás amit meglátsz a gépen/telón
- **Megjelenítési:**
 - Adatkonverzió
 - Protokollok: adatformátumok és transzformációs szabályok
- **Ülés:**
 - Kapcsolat menedzsment
 - Szinkronizációs pont menedzsment
- **Szállítói:**
 - (De)Multiplexálás
 - Sorrendhelyes továbbítás, üzenet küldése egy célállomásnak
 - Protokollok: portszám, hibajavítás, folyamfelügyelet
- **Hálózati:**
 - Csomagtovábbítás, útvonalválasztás
 - Csomagok ütemezése, pufferezés
 - Protokollok: routing táblák karbantartása, globálisan egyedi címzés
- **Adatkapcsolati:**
 - Adatok tördelése, csomaggá alakítás
 - Per-hop megbízhatóság, folyamvezérlés
 - Protokollok: fizikai címzés (MAC/IB address)
- **Fizikai:**
 - Információ átvitele két fizikailag összekötött eszköz között
 - Specifikálja egy bit átvitelét
 - Protokollok: bit kódolásának sémája

3. előadás (Physical)

2024. december 18., szerda 11:53

Fizikai réteg:

- **0-k és 1-esek:** van vagy nincs feszültség/áramerősség
- **Fourier-sor** felhasználása:
 - PI karakterek kódja (digitális jel) nem periodikus
 - De elképzelhető, hogy **végtelemszer ismétlődik**
- Elnyelődés (*attenuation*):
 - Küldési és vételi energiák hányadosa
 - Mértékegység: decibel
 - Befolyásoló tényezők: közeg, távolság
- **Valódi közegben hátráltató tényezők:**
 - Frekvenciafüggő elnyelődés
 - Fáziseltolódás
 - Zaj
- **Sávzsélesség:**
 - Az a **frekvenciatartomány**, ahol a **csillapítás nem túl nagy**
- **Vezeték nélküli átvitel:**
 - **Frekvencia:** elektromágneses hullám rezgésszáma / sec
 - Jele: f
 - Mértékegység: Hertz (Hz)
 - **Hullámhossz:** két egymást követő hullámvölgy/csúcs közötti táv
 - Jele: λ
 - **Fénysebesség:** $c = 3 \cdot 10^{23} \frac{m}{s}$
 - $\lambda \cdot f = c$
- **Műholdas kommunikáció:**
 - Transzponderek alkalmazása
 - Kapott jelek felerősítése, továbbítása más frekvencián

Adatátvitel:

- **Szinkronizációs probléma**
 - Órajel szinkronizálás
 - Hogyan különböztessük meg a 0 értéket a nincs jeltől?
 - Elcsúszás probléma
- **Felügyelet szükséges** a szinkronizációhoz
 - Explicit órajel
 - Párhuzamos átviteli csatornák
 - Rövid átvitel esetén
 - Kritikus időpontok
 - Csak szimbólom/blokk kezdetén szinkronizálunk
 - Feltételelesen rövid ideig szinkronban futnak
 - Szimbólum kódok
 - Önütemező jel
 - Szignál tartalmazza az információt

Multiplexing:

- Fizikai csatorna logikailag több alcsatornára bomlik = több jel átvitele egyszerre
- Szükséges eszköz: multiplexer
- **Térbeli / Space-Division Multiplexing (SDM):**
 - Minden csatornához külön vezeték / antenna
- **Frekvencia / Frequency-Division (FDM):**
 - Minden jel más frekvencián érkezik
- **Hullámhossz / Wavelength**
- **Időbeli / Time**

Code Division Multiple Access (CDMA):

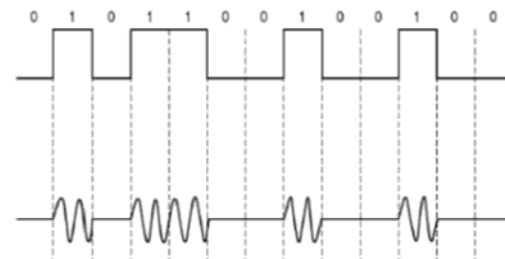
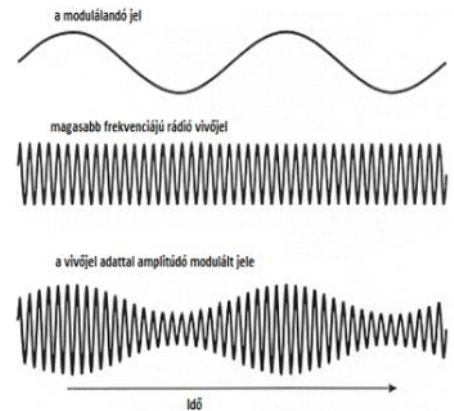
- Minden állomás egyszerre sugározhat a rendelkezésre álló frekvenciasávon
- **Algoritmus:**
 - Bitidőt m darab intervallumra osztjuk
 - Minden állomáshoz tartozik egy m bites kód (töredéksorozat)
 - Ha **0-t** továbbítok: elküldöm a saját **töredéksorozat**om
 - Ha **1-et**: annak az **1-es komplementjét**

Átviteli közegek:

- **Vezetékes**
 - Mágneses adathordozók
 - Sodort érpár
 - Koax kábel
- **Vezeték nélküli:**
 - Rádiófrekvenciás
 - Mikrohullámú
 - Infravörös
 - Látható fényhullámú
- **Műhold típusok:**
 - Geoszinkron
 - Közepes és alacsony röppályás

Alap és szélessávú jelátvitel:

- **Alapsáv (baseband)**
 - Digitális jel árammá/feszültséggé alakul
 - Minden frekvencián átvitel
- **Szélessáv (broadband)**
 - Széles frekvenciatartományban van jelátvitel
 - Jel modulálása:
 - Amplitúdó moduláció
 - Frekvencia moduláció
 - Különböző vívőhullámok egyidejűleg való használata
 - **Amplitúdó moduláció:**
 - **Analóg:** Vivőjel amplitúdóját illesztet a modulálandó jelre
 - **Digitális:** Amplitúdó méretéből szűrőd le a jelet
 - **Frekvencia moduláció:**
 - **Analóg:** Vivőjel frekvenciáját a modulálandó alapján változtatjuk
 - **Digitális:** frekvencia változásból szűrőd ki a jelet
 - **Fázis moduláció: (nem nagyon használt)**
 - **Analóg:** ugyanaz, jel amplitúdója befolyásolja a vivő fázisát
 - **Digitális:** ugyanaz, vivő fázisváltozásából szűrész le infót



4. előadás (Data-Link)

2024. december 18., szerda 14:56

Keretezés:

- Adatblokk kódolása
- Hibakeresés, és esetlegesen javítás



- Bájt beszúrás:**
 - FLAG bájt: adat eleje + vége
 - ESC: biztosítja, hogy helyesen megtaláljuk a FLAG-et
- Bit beszúrás:**
 - Speciális bitmintával kezdünk és végzünk
 - Minden fix bit-részsorozat után beszúrunk egy jelző bitet
 - Ha nem lehetséges mintát fogadunk: eldobjuk a keretet
- Hibák:
 - Egyszerű bithiba: 1-ről 0-ra vagy 0-ról 1-re változik
 - Csoportos hiba:
 - Olyan részsorozat, ahol az első és utolsó hibás
 - Nem létezik olyan m hosszú sorozat, ami helyes
 - Ilyenkor m a védelmi övezet
- Hibadetektálás:
 - Duplikálás = szar
 - Paritásbit
 - Egyesek száma páros legyen
 - Valamennyire szar
 - Hibajavító kódok: kevésbé megbízható helyeken célszerű
 - Detektálás és újraküldés: megbízható helyen jobb
- Szükséges a redundancia:**
 - Adatbitek: m
 - Ellenőrző bitek (adatból származt): r
 - Teljes méret = m + r
- Hamming távolság:** d(x, y) ahol x és y a kódszavak
 - Egy S halmaznál: d(S) = min_{x,y in S} d(x, y)
 - Ha d(S) = 1: nincs hibafelismerés
 - Ha d(S) = 2: a hiba felismerhető, de nem javítható

Tétel
Minden C ⊆ {0,1}ⁿ kód, ahol d(C) = k (∈ ℕ₊). Akkor teljesül az alábbi összefüggés:

$$|C| \sum_{i=0}^{k-1} \binom{n}{i} \leq 2^n$$

Forgalomszabályozás:

- Visszacsatolás alapú (feedback-based flow control)
- Sebesség alapú (rate-based f.c.)

Paritás bit használata:

- Kódszó bitek megszámozása 1-től
- 2 hatványai ellenőrzőbitek
- A k-adik pozíciót a k értékének megfelelő 2 hatványokkal ellenőrizzük
 - k = 13 = 1 + 4 + 8, az 1., 4. és 8. bit fogja ellenőrizni

ASCII karakter	ASCII decimális	Üzenet forrás bitjei	Az előállt kódszavak
E	69	1000101	10100000101
L	76	1001100	10110011100
T	84	1010100	00110101100
E	69	1000101	10100000101
	32	0100000	10001100000
I	73	1001001	11110011001
K	75	1001011	00110010011

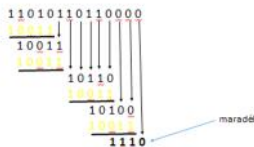
Hibák felismerése és javítása Hamming-távolsággal:

- d hibás bit felismeréséhez: d(S) ≥ d + 1
- d hibás bit javításához: d(S) ≥ 2d + 1

Hibajelző kódok:

- CRC (ciklikus redundancia)
 - A kódot ℤ₂ alatti polinomnak értjük
 - Kivonás és összeadás logikai XOR művelet
 - Az algoritmus:
 - G(x) - fok r - generálópolinom, mindkét fél ismeri
 - G(x)-hez fűzzük hozzá r db 0 bitet [G'(x)]
 - G'(x) / G(x) mod 2
 - Az osztás maradékát vonjuk ki G'(x)-ből, ennek eredménye a továbbbitandó keret + ellenőrző összeg [T(x)]
 - Vevő megkapja T(x) + E(x)-et - E(x) a hibapolinom -, és ha E(x) / G(x) nem 0, akkor hiba van

Keret: 1101011011
Generátor: 10011
A továbbbitandó üzenet: 1101011011110



Adatkapcsolati protokollok:

- Szimplex**
 - Egy irányú kommunikáció
 - Korlátozás nélküli:
 - Nincs nyugta/sorszámozás
 - Folyamatosan küld
 - Stop-and-wait:
 - Küld, nyugtát vár, ismétel
 - Fél duplex!
 - Kiegészítve számlálóval: ha nem jön válasz, újraküldi
- Fél-duplex**
 - Egyszerre egy irányú kommunikáció
- Duplex**
 - Mindkét irányba lehet egyszerre kommunikálni
- ABP (Alternáló bit protokoll):**
 - Egyesével küldjük, nullás sorszámmal kezdve
 - Fogadásnál ellenőrzünk sorszámot és az összeget,
 - Ha hiba eldobjuk és nem nyugtázunk
 - Ha jön nyugta: sorszám növelése mod 2
 - Ha nem jön nyugta akkor újraküldünk

$$\begin{array}{r} 11110000 \\ - 10100110 \\ \hline 01010110 \\ + 10011011 \\ \hline + 11001010 \\ \hline 01010001 \end{array}$$

5. előadás (Data-Link)

2024. december 18., szerda 18:23

Csúszó-ablak protokoll:

- A küldő **max n**-et küldhet (adási/vételi ablak)
- Fogadó n keretnek **megfelelő puffert** allokál
- Sorozatbeli **pozíció** adja a keret **sorszámát**
- Hibákat **eldobjuk**
- **Nyugta** tartalmazza a **sorszámot**
- **Pipelining:**
 - Nagy RTT kihatással lehet a **sávszélesség kihasználtságára**
 - Adatsebesség és RTT szorzata nagy: **nagyméretű ablakok** használata (pipelining)
 - **Visszalépés n-el**
 - Összes keret újraküldése a hibásig
 - Nagyon pazarló nagy hibaarány esetén
 - **Szelektív ismétlés:**
 - Hibákat eldobása, többi puffelése
 - Esetleg negatív nyugta használata
 - Nagy memóriaigény

Közeg Hozzáférés Vezérlés / Media Access Control:

- Szabályok definiálása **közeg megosztásához**
- Stratégiák **ütközések elkerüléséhez**
- **Ki férjen hozzá épp az adatszóró csatornához?**
 - **Statikus csatornakiosztás:**
 - **Frekvenciák felosztása** felhasználók között
 - Előnyös ha fix felhasználó van nagy adatforgalomigénnyel
 - **Időalapú felosztás**
 - Egyik **se hatékony burst**-ök esetén
 - **Dinamikus kiosztás**
 - Verseny(mentes), avagy korlátozott verseny protokollok
 - **Hatékonyság:**
 - Átvitel, Késleltetés, Fairness (*mindenki egyenrangú-e*), Terhelés
- **ALOHA algoritmus**
 - Azonnal küldünk
 - Mindig nyugtázunk
 - Ütközésnél random ideig várunk, és újraküldjük
 - **Poisson folyam** **[már a név miatt sem tetszik :)]**
 - Érkezés valószínűsége egy t intervallum alatt arányos t hosszával
 - Nem függ t kezdetétől
 - Több érkezés esélye t alatt közelít a 0-hoz
 - **Réselt ALOHA:**
 - A csatornát időrésekre osztjuk, csak rések határán küldhetünk

CSMA protokollok:

- **1-perzisztens:**
 - Vivőjel érzékelés: **minden állomás belehallgat**
 - Leadás előtt **hallgatunk, hogy üres-e** a csatorna, ha igen, **azonnal küldünk**
 - **Ütközéskor random ideig várunk**, majd újraküldünk
- **Nem-perzisztens:**
 - Ha hallgatunk és **foglalt**, akkor **random ideig várunk**, majd újra hallgatunk
 - Amúgy ugyanaz
- **P-perzisztens:**
 - Ha nem foglalt, **p eséllyel küldünk**, vagy 1-p eséllyel **visszalépünk** és várunk
 - Ez addig megy ameddig el nem küldtük a keretet, vagy **más el nem kezdett küldeni**, ilyenkor **ütközés**

6. előadás (Data-Link)

2024. december 18., szerda 19:15

CSMA

- **CD változat (ütközés detektálás)**
 - Küldés közben **figyeljük a csatornát**
 - Ütközésnél **megállunk**, random ideig **várakozunk**, **újrakezdjük**

Ütközéskor "jam" jel küldése

Binary Exponential Backoff

- $k \in [0, 2^n - 1]$, $n \leq 10$ egyenletes eloszlás szerint, ahol n az ütközések száma
- Várunk k időegységet az újraküldésig
- 16 újrapróbálkozásnál eldobjuk a keretet

Minimális keretméret: 64 bájt (az állomásoknak elég idő kell az ütközés detektálásához)

Kábel minimális hossza: $Táv = \min(\text{keretméret}) * \text{fénysebesség} / (2 * \text{ráta})$

Maximális keretméret (Maximum Transmission Unit): 1500 bájt

- Túl hosszú csomaggal nagy javítási költség
- Több bájtba kerül a fejléc, nagyobb feldolgozási idő

Ütközésmentes protokollok:

- N állomás (1- N -ig sorszámozva)
- Réselt időmodell
- **Bittérkép protokoll:**
 - Ütköztetési periódus **N db időrés**
 - Ha **i . állomás** küldene, akkor **i . időrésben** küld egy **1-es bitet** (*broadcast*)
 - Végén **minden állomás** ismeri a küldőket
- **Bináris visszaszámlálás:**
 - Minden állomás azonos hosszú **bináris azonosítóval** rendelkezik
 - Küldő állomás **bitenként elküldi az azonosítóját**
 - Ezek **OR kapcsolatba** kerülnek
 - Ha **0-t** küldök, **de egyet hallok vissza**, akkor **visszalépek**, mert valaki már küld
 - **Mok és Ward:**
 - Sikeres átvitel után permutáljuk a címeket

Korlátozott versenyes protokollok

- Ötvözni a versenyhelyzetes és ütközésmentes protokollokat
- **Adaptív:**
 - **Kis terhelés:** versenyhelyzetes
 - **Nagy terhelés:** ütközésmentes
- **Adaptív fabejárási protokoll:**
 - 0. pillanatban **mindenki küldhet** (ütközésnél mélységi bejárás)
 - Rések a **fa** egyes **csomópontjai**
 - **Ütközéskor** a csomópont bal és jobb **részfáján folytatódik a keresés**
 - Ha egy **bitrés kihasználatlan** vagy pontosan **egy állomás küld**, akkor **vége a keresésnek**

Bridging / bridge-ek:

- LAN-ok összekapcsolása
- **Előnyök:**
 - Korlátozzák az ütközési tartományok méretét
 - Skálázhatóság növelése
- **Hátrányok:**
 - Komplexebb eszközök
- **Cél:** teljes átlátszóság / **plug-and-play**, **önmagát konfiguráló**
- **Továbbító tábla** készítése, **címek tanulása** (küldésekkor források címei)
- **Hurkok problémája: feszítőfa** készítése és karbantartása (802.1)
 - Egyik bridge a **gyökér**
 - **Ennek meghatározása:**
 - BPDU üzenetek
 - Switchek ez alapján választják ki:
 - ◆ Gyökér elemet
 - ◆ Gyökér portot
 - ◆ Következő bridge-et a gyökér felé
 - Ehhez mind megkeresi a **legrövidebb utat**
 - Ennek **uniója** a feszítőfa

A switchek a bridge egy speciális esetei, minden port egy hoszthoz kapcsolódik

7. előadás (Network)

2024. december 19., csütörtök 15:39

Forgalomirányító algoritmusok:

- **Elvárások:** igazságos, optimális
- Osztályok:
 - **Adaptív:** topológia és forgalom is befolyásolhatja a döntés
 - **Nem-adaptív:** offline meghatározás
- **Optimalitási elv:**
 - Ha J router az I-től K-ig menő optimális úton helyezkedik el, akkor J-től K-ig is ugyanez az optimális út
 - Egy célba tartó optimális utak fája: **nyelőfa**
- **Legrövidebb út alapú forgalomirányítás:**
 - Routerok és linkek **súlyozott gráfot** alkotnak
 - Linkek súlyai alapján megkeresni a **legkisebb súllyal rendelkező utat**
- **Adaptív algoritmus alosztályok:**
 - **Távolságvektor alapú**
 - Minden routernek **saját táblázata** van minden célhoz ismert **legrövidebb távolságról** és ennek azonosítójáról
 - **Szomszédoktól kapott információk** alapján frissítik
 - **RIP** (Routing Information Protocol)
 - **Kapcsolatállapot alapú**
 - Szomszédok **felkutatása** (**HELLO**) és **késleltetés/költség** (**ECHO**) mérése
 - Ezekből **csomag összeállítás** és elküldés az **összes többi routernek**
 - Minden router **kiszámolja a legrövidebb utat** (**Dijkstra**)
 - **OSPF:** cégek (**átfedő területek**)
 - **IS-IS:** internetszolgáltatók (**hierarchikus**)

Forgalomirányítás:

- **Unicast:**
 - Forrás és cél egyedi IP azonosítóval rendelkezik
- **Broadcast:**
 - Külön csomag mindenkinek: pazarló
 - **Elárasztás:** kétpontos kommunikációhoz nem megfelelő
 - **Többcélú routing:** csomagban egy lista minden célról, router minden megfelelő irányba küld egy másolatot
 - **Nyelőfával:** minden irányba továbbküldjük, amik a feszítőfához tartoznak, kivéve azt, amin jött
 - **Visszairányú továbbítás:** ha csomag érkezett, megnézzük hogy onnan jött-e, amerre általában mi szoktuk-e ezeket küldeni, ha igen, akkor tovább küldöm minden irányba, ha nem, eldobom
- **Multicast:**
 - Csoportkezelés szükséges
 - Kiszámítunk egy olyan feszítőfát, ahol az ágak a csoporton belüli tagokhoz vezetnek, csak erre továbbítunk

Címzési struktúrák:

- **Sík (Flat):**
 - **Minden hosztot** egy egyedi **48 bites MAC címmel** azonosítunk
 - Minden címhez kéne egy bejegyzés a routing táblába
 - Túl nagy, lassú, nehéz karbantartani
- **Hierarchikus**
 - **Szegmensekre osztható címek**
 - Minden szegmens egy réteget ír le

IP címzés:

- **4 bájtos címek**
- Több évtizedig osztályos
 - A: 0.0.0.0- (első bájt hálózat)
 - B: 128.0.0.0- (első 2)
 - C: 192.0.0.0- (első 3)
 - D: 224.0.0.0-
 - E: 240.0.0.0-
- Minden hálózat **utolsó kiosztható címe a broadcast cím**
- **Alhálózatok használata**
 - **Kívülről** egynek néz ki
 - **Belülről** többre osztható
 - **Alhálózati maszk** használata (routernek ismernie kell)
- **CIDR címzés**
 - Címek elfogytak
 - **Osztályok nélküli környezetek** és közöttük forgalomirányítás
 - Forgalomirányítás bonyolódik
 - Túl sok bejegyzés: **csoportos bejegyzések**
 - Minden bejegyzéshez ip, alhálózati maszk, kimeneti vonal

NAT:

- Minden belső hálózathoz **egy vagy kevés IP** cím kerül kiosztásra
- Hálózaton **belül egyedi címek** használata
- **Címtartományok:**
 - 10.0.0.0/8
 - 172.16.0.0/12
 - 192.168.0.0/16
- **Port mezők** használata mutatók tárolására
- NAT boxban **fordítási tábla**

8. előadás (Network)

2024. december 19., csütörtök 19:03

Hierarchikus forgalomirányítás:

- Indok: nagyobb hálózathál nagyobb a routerek erőforrás igénye
- **Többszintű hierarchia alkalmazása**
 - N router esetén $\ln(N)$ szint
- **Gerinchálózatok**
 - Ezekhez területi / regionális kiszolgálók csatlakoznak
- IP protokoll
- **IP-fragmentáció:**
 - Minden hálózathál megvan az **MTU**-ja (*Maximum Transmission Unit*)
 - **Szétvágjuk**, ahol **kisebb MTU** szükséges, majd a fogadó összerakja
 - **Költséges**, ha lehet, elkerülni
 - **MTU felderítő protokoll:**
 - **DON'T FRAGMENT** bittel csomagküldés
 - Addig küldünk, **amíg egy meg nem érkezik**
 - **CAN'T FRAGMENT hiba routertől**, tartalmazza a hálózat **MTU méretét**
 - Darabok kezelését speciális hardver végzi

IPv6:

- IPv4 címek fogynak / elfogytak
- **128 bites címek (16 bájt)**
- Átalakult fejléc, számos mező hiányzik
- **Teljesítmény növekedés:**
 - Nincs checksum
 - Nincs fragmentáció kezelés
 - Egyszerű routing tábla szerkezet
 - Egyszerű auto-konfig (Neighbour Discovery Protocol)
- **Esetleges lehetőségek**
 - Mobil IP
 - Forrás routing
 - Privacy kiterjesztés
- Átmenetben az **internet magja** a probléma
 - Használunk **tunneleket** IPv6 csomagok becsomagolására :DDD

Internet forgalomirányítás:

- **2 szintű hierarchia:**
 - **Első szint:** autonóm rendszerek (**AS**)
 - Intra-domain protokollok
 - RIP, OSPF
 - **Második szint:**
 - Inter-domain protokollok
 - BGP
- **Miértttt?**
 - Routing algoritmusok **nem elég hatékonyak**, hogy a teljes interneten működjenek
 - Szervezetek **más-más politika** alapján vezérelnek
 - Szervezetek **elrejtetik** a belső hálózatok szerkezetét / **eldöntések**, merre akarnak forgalmazni
- **ASN:** AS számok, azonosítók (16 vagy 32 bites)
- **Inter-domain routing:**
 - Minden AS-nek ugyanazt kell használnia, szemben az Intra-domainnel
 - Rugalmas, skálázható
- **Border Gateway Protocol:**
 - Eszerint **csak AS-ek** és közöttük **kapcsolatok** vannak
 - Politikai szabályok támogatása (ne menjen erre arra stb)
 - **Hálózat típusok:**
 - Csonka: egy összeköttetés
 - Többszörösen bekötött: használható átmenő forgalomhoz, de letiltja
 - Transzit: megkötéssel / fizetségért átengedi a forgalmat
 - **Útvonalak szomszédoktól**
 - **Pontozzuk** az útvonalakat, végtelen érték a szabályoknak nem megfelelőeknek
- **Útvonal vektor protokoll (PVP):**
 - Hurkok / körök detektálása
 - Routing a leghosszabb megegyező prefix alapján

9. előadás (Network, Transport)

2024. december 19., csütörtök 19:58

Internet Message Control Protocol:

- Váratlan események kezelése / jelentése

Address Resolution Protocol:

- IP cím megfeleltetése MAC (fizikai) címnek
- Broadcast üzenet segítségével, a válasz tartalmazza a címet

Dynamic Host Configuration Protocol:

- Hálózati azonosítók automatikus kiosztása a hostoknak
- IP címet pl tud osztani MAC cím alapján, vagy teljesen dinamikusan
- Ezeket a címeket **bérbe adja** (*lease time*)
 - Ez meghosszabítható vagy idő előtt lemondható

Virtual Personal Networks:

- A internettől **logikailag elkülönülő** magánhálózat
- "Fizikailag távoli hostok logikailag egy hálózatot alkotnak"
- Hostok egymás **normál módon eléri**k
- **Virtuális linkek** alagutak segítségével
- **IPSEC:**
 - IP réteg biztonságossá tétele
 - Bizalmas, autentikált

Szállítói réteg:

• User Datagram Protocol:

- 8 bájtos fejléc
- **Forrás és célportszámok** használata (demultiplexálás)
- Hiba **detektálása csak**, semmi más
- **Kapcsolatmentes**

• Transmission Control Protocol:

- **Megbízható, sorrendhelyes**
- **Kétirányú** kommunikáció
- 20 bájtos fejléc
- **SYN, ACK, FIN** flagek kapcsolatépítéshez (SYN flood = DoS)
- Transzferelt bájtfolymatot **szegmensekre** bontjuk
- **Minden bájt számozott** (32 bites érték, körbeér idő után, random kezdés)
- **Csúszóablak:**
 - Túl sok csomag: túlterheli a vevőt
 - Vevő átküldi a **buffer méretét (n)**
 - **Maximum ekkora (n)** méretű üzenet küldhető **ACK nélkül**
- **Nyugták:**
 - Minden csomaghoz
 - Kumulált (n sorszám minden $k < n$ csomagot nyugtáz)
 - Negatív nyugta (NACK, nem megérkezett csomag jelzése)
 - Szelektív (SACK, melyik érkezett meg, nem feltétlen sorrendben)

10/13. előadás (Application)

2024. december 19., csütörtök 21:35

DNS (Domain Name System)

- Elosztott adatbázis alapú
- Hierarchikus névtér (.com -> google.com -> mail.google.com)
 - Egy fát képez
 - Max mélység 128
 - Minden domain név egy részfa
- Nincs névutközés
- Zónákra bomlik a fa
 - Mindegyik zónáért egy szervezet felel
- Egy szerver **csak a zónájához** tartozó hostok és domainek rekordjait tárolja
- Ismeri a **Root szerver címét**
 - Root minden TLD-t ismer
- **Top Level Domains:**
 - 22+ általános (.com, .edu, .gov)
 - Kb 250 államoknak
- Minden ISP-nél / cégnél **lokális névszerverek**, itt kezdődik egy lekérdezés
 - **Iteratív lekérdezés:** *lokális adja a választ, vagy hogy kinél kell keresni*
 - Jó ha magas a szerver terhelése
 - **Rekurzív:** *lokális keresi ki és adja a választ*
 - Jobb teljesítmény kliens szempontból
- Könnyű mozgatni szervert: rekordban IP cím kicserélése
- **Aliasing:** egy gépnek több neve lehet
- **Load Balancing:** egy névhez több szerver is tartozhat
- DNSSEC: kritikus rekordok kriptografikus aláírással biztosítva

HTTP:

- Kliens-szerver model
- **Statikus weboldal:**
 - Nem változik, csak kézi átírásra
- **Dinamikus weboldal:**
 - JavaScript, PHP
- **Persistent / Non-persistent HTTP:**
 - Egyszerre több / egy objektum küldhető egy TCP kapcsolattal
- **Request és Response üzenetek**
 - ASCII formátum
 - GET, POST, HEAD
- Feltöltés POST-tal vagy URL segítségével
- Cookie-k (~~teszaram~~)

11. előadás (Transport)

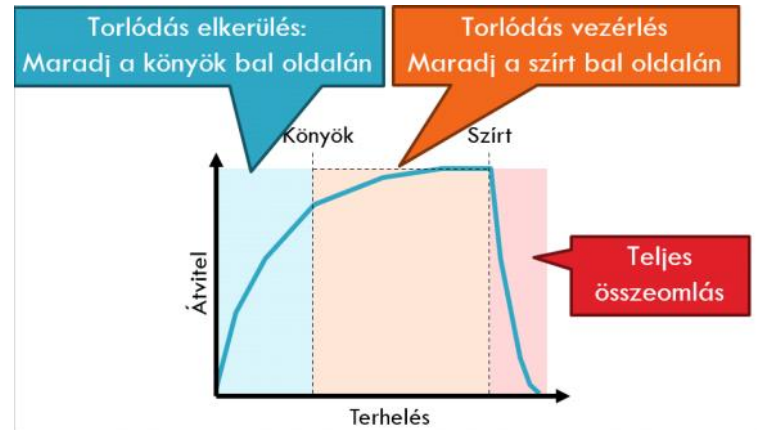
2024. december 19., csütörtök 22:08

Hiba detektálás:

- **Kontrollösszeg** detektálja
- **Sorszámokkal** helyes átviteli sorrend
 - Duplikáltakat eldobni
 - Hiányzó sorszám = elveszett csomag
- **Küldő oldalon: elveszett csomag = nincs nyugta**
 - Timeout használata
- **RTT: Round Trip Time**

Torlódás:

- **Nagyobb a terhelés, mint a hálózat kapacitása**
- Csomagvesztés, elvesztegetett sávszélesség
- **Advertised Window:**
 - Csak a **fogadót védi**
 - Legyen **állítható**
- **Megoldások:**
 - Erőforrás foglалás
 - Dinamikus beállítás
 - Próbák használata
- **TCP torlódásvezérlés:**
 - Minden kapcsolat rendelkezik **ablakkal**
 - Detektálás **nyugtával** (vagy hiányával)
 - Ráta beállítás
 - **Cwnd** növelése/csökkentése nyugta esetén/hiányában
 - **Lassú indulással** (additive increase)
 - **Reno**: gyors újraküldés



12. előadás

2024. december 19., csütörtök 22:38

Ha ebből akármit is kérdez bassza meg