



AlexAegis asd



1 contributor

Raw

Blame



1512 lines (858 sloc) 58.1 KB

# Definitions

## 1. lecture

- Mi a hálózati host?

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

- 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.

- Definiálja a **propagációs késést**.

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

Jelölése:  $d_{\text{prop}}$  vagy  $d$

- Definiálja az átviteli késleltetés.

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

Jelölése:  $d_T$

- Definiálja a **jel sávszélességet**.

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

Tipikusan Hertz-ben mérik.

- Definiálja a **hálózati sávszélességet**.

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

- 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)

- Sorolja fel a **hálózati kiterjedéseket**.

**PAN:** Personal Area Network (1 m<sup>2</sup>)

**LAN:** Personal Area Network (10-1000 m<sup>2</sup>)

**MAN:** Metropolitan Area Network (10 km<sup>2</sup>)

**WAN:** Wide Area Network (100-1000 km<sup>2</sup>, de az internet is)

- 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.

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

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

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

Rendszerfüggetlenség

Nincs központi felügyelet

LAN-okból áll

Globális

Szolgáltatásokat nyújt, pl WWW, e-mail, fájlátvitel

- 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

- 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

- 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)

- 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
- 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 címzés (pl. MAC address, IB address)
  - Példák
    - Ethernet

- Wifi
- InfiniBand
- 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
- 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
- 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
- Mi a feladata és mik a főbb funkcionálisai az ISO/OSI modell **alkalmazási rétegének**?
  - Szolgáltatás
    - Bármilyen...
  - Interfész
    - Bármilyen...
  - Protokoll
    - Bármilyen...
  - Példák
    - kapcsolod be a mobilod és nézd meg milyen appok vannak rajta...

## 2. lecture

---

- 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

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

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

- 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.

- 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?

- 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

- 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

- Mit mond ki a **Nyquist tétel**?

*Zajmentes csatorna* esetén a

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

**H:** sávszél

**V:** szimbólumok száma

- 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

- 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** (*egyenestáv mentén terjed, elhalkulás problémája, olcsó*)

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

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

**Műholdas**

- Mit nevezünk **frekvenciának**? Hogyan jelölik? Mi a mértékegysége?

Az elektromágneses hullám másodpercenkénti rezgésszáma.

Jelölése:  $f$

Mértékegysége: Hertz (Hz)

- Mi a **hullámhossz**?

Két egymást követő hullámcsúcs (vagy hullámvölgy) közti távolság

Jelölése:  $\lambda$

- Mi a **fénysebesség**?

Elektromágneses hullámok terjedési sebessége vákuumban

Jelölése:  $c = 3 \cdot 10^8$  m/s

*rézben és üvegszálaban ennek csak 2/3-a*

- Összefüggés **hullámhossz, frekvencia és fénysebesség** között

**hullámhossz  $\times$  frekvencia = fénysebesség**

- Soroljon fel 3 **elektromágneses tartományt** a frekvenciáik növekvő sorrendjében!

i. Rádió

ii. Mikrohullám

iii. Infravörös

iv. Látható



- v. Ultraibolya
- vi. Röntgensugár
- vii. Gammasugár

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

Közeg	Frekvenciatartomány
Sodort érpár	$10^4 - 10^6$ Hz
Koax kábel	$10^5 - 10^8$ Hz
Optikai szál	$10^{14} - 10^{15}$ 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

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

Bit	Jel
1	Magas
0	Alacsony

**Deszinkronizációra hajlamos**

- 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	_	_	_	_	_

- 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	_	_	_	_	_	_	_	_	_	_

- 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

- Mik a főbb tulajdonságai a **baseband** (alapsávú) átvitelnek?

a digitális jel direkt árammá vagy feszültséggé alakul

a jel minden frekvencián átvitelre kerül

átviteli korlátok

- Ismertesse a digitális **alapsávú átvitel** struktúráját!

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

- 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**

- Ismertesse a digitális **szélessávú átvitel** struktúráját!

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

- Mi az **amplitúdó moduláció**?

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

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

$t$ : periódus idő

$f$ : frekvencia

$s(t)$ : amplitúdó

$\varphi$ : eltolás

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

- Mi a **frekvencia moduláció**?

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

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

$t$ : periódus idő

$s(t)$ : frekvencia

$a$ : amplitúdó

$\varphi$ : eltolás

- Mi a fázis moduláció?

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

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

$t$ : periódus idő

$f$ : frekvencia

$a$ : amplitúdó

$s(t)$ : eltolás

### 3. lecture

---

- Ismertesse a **médium többszörös használatának 5 módszerét!**

- **SDM - Térbeli** multiplexálás (**Space-Division Multiplexing**)

Külön vezeték vagy antenna

- **FDM - Frekvencia** multiplexálás (**Frequency-Division Multiplexing**)

Több szignál kombinációja adja az átvitelt, minden szignálhoz más frekvencia tartozik

- **WDM - Hullámhossz** multiplexálás (**Wavelength-Division Multiplexing**)

Optikai kábeleknél használt

- **TDM - Időbeli** multiplexálás (**Time-Division Multiplexing**)

Jelsorozat időintervallumokra szegmentálása, minden állomás saját időszeletet kap

- **CDMA - Kód** multiplexálás (**Code Divison Multiple Access**)

Állomások egyfolytában sugározhatnak a teljes frekvenciasávon

Feltételezi, hogy a többszörös jelek lineárisan összeadódnak

**Kulcs:** a hasznos jel kiszűrése

- 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 tulajdonságát a multiplexáláshoz.

Algoritmus:

- i. Minden bitidőt  $m$  darab intervallumra osztunk (chip)
- ii. Minden állomásnak van egy  $m$  bites kódja (chip sequence - páronként ortogonálisak)
- iii. 1-es bitet akar továbbítani egy állomás: saját töredéksorozatát küldi tovább
- iv. 0-ás bitet akar továbbítani egy állomás: saját töredéksorozatát 1-es komplementjét küldi tovább

- 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.

- Hogyan áll elő a  $H(2^k)$ -nal jelölt Walsh mátrix?

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

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

Ahol

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

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

- 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)

- Milyen módszereket ismer a **keretezésre** az adatkapcsolati rétegben?

**Bájt alapú:** karakterszámlálás, bájtbeszúrás

**Bit alapú:** bitbeszúrás

**Óra alapú:** SONET - (Synchronous Optical Network) kódolás

- Hogyan működik a **karakterszámlálás**?

Keret fejlécében megadjuk a keretben lévő karakterek számát

Érzékeny a hibára

- 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

- 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

- 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

- 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

- Mi az **egyszerű bithiba** definíciója?

Az adat **1** bitje **1** helyett **0** lesz vagy fordítva

- 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)

- Definiálja egy tetszőleges  $S$  kódkönyv **Hamming távolságát**?

$S$  kódkönyvben szerepeljenek egyenlő hosszú bitszavak, ekkor  $S$  Hamming-távolsága:

$$d(S) = \min \{ d(x,y) \mid x \neq y \text{ eleme } S \}$$

- Mi az a **Hamming korlát**?

$C$ : kód ( $n$  hosszú szavakból)

$$|C| \cdot \sum_{i=0}^{d(C)-1} \binom{n}{i} \leq 2^n$$

- 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

- Milyen összefüggés ismeretes egy tetszőleges kódkönyv, a Hamming távolsága és **hibafelismerő** képessége között?

$d$  bithiba felismeréséhez a kódkönyv H-távolsága minimum  $d+1$  legyen

- Milyen összefüggés ismeretes egy tetszőleges kódkönyv, a Hamming távolsága és **hibajavítási** képessége között?

$d$  bithiba javításához a kódkönyv H-távolsága minimum  $2d+1$  legyen

- Mikor érdemes hibajelző kódot és mikor hibajavító kódot használni?**

**Hibajelző:** megbízható hálózat (ARQ) - olcsóbb

**Hibajavító:** megbízhatatlan hálózat, gyakori hibákkal (FEC) - sok ismétlés elkerülésére

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

Egy paritást használó módszer

i. A kódszó bitjeit számozzuk meg 1-gyel kezdődően

ii. 2 egészhatvány sorszámú pozíciói lesznek az ellenőrző bitek, azaz 1,2,4,8,16...

iii. A maradék helyeket az üzenet bitjeivel töltjük fel

iv. Mindegyik ellenőrző bit a bitek valamilyen csoportjának a paritását állítja be párosra (vagy páratlanra)

v. 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

- 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

- Mi a **CRC**? Mire használható?

Cyclic Redundancy Check, **hibajelző kód**

Bitsorozatokat  $\mathbb{Z}_2$  feletti polinomok reprezentációjának tekinti

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

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

- Mikor **nem ismeri fel** a hibát a vevő oldal?

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



- **CRC** esetén mit lehet mondani **hibajelző képességéről**, ha a generátor polinom  $x+1$  többszöröse?
  - ez esetben minden páratlan számú hiba felismerhető
- 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
- 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)
- Mutassa be röviden a **szimplex protokollt zajos csatorna** esetén!
  - Környezet:
    - Adó-vevő mindig kész

Van feldolgozási idő

Végtelen puffer

A csatorna hibázhat

o 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)

• Mit nevezünk **adási ablaknak**?

A küldhető sorozatszámok halmaza

• Mit nevezünk **vételi ablaknak**?

A fogadható sorozatszámok halmaza

• 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

• Mi a **visszalépés N-nel stratégia** lényege?

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

• 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, egyenlő nagyobb vételi ablak kell

- 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 hálózati réteg opcióinak megbeszélésre, amely független az alkalmazott hálózati réteg protokolltól.
- 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)
- Röviden mutassa be a **frekvenciaosztásos nyalábolás** módszerét!
  - N db userhez a sáv szélt N egyenlő méretű sávra osztja
  - Fix számú usernél, nagy forgalomigénynél jó
  - Löketszerű forgalom esetén problémás
- Röviden mutassa be az **időosztásos nyalábolás** módszerét!
  - N db userhez az időegységet N egyenlő méretű időrésre osztja
  - Löketszerűnél nem jó
- A csatorna modellben mit nevezünk **ütközésnek**?

Ha két keret egy időben kerül átvitelre, akkor átlapolódnak, és értelmezhetetlenné válnak
- Írja le a **folytonos és a diszkrét időmodell** lényegét!
  - **Folytonos**

Mindegyik állomás tetszőleges időpontban megkezdheti a kész keretének sugárzását
  - **Diszkrét**

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

- 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

- Hogyan működik az egyszerű **ALOHA** protokoll?

Ha van küldendő adat, akkor a host elküldi.

- Mit jelent a **keretidő** az **ALOHA** protokoll esetén?

Egy fix hosszúságú keret átviteléhez szükséges idő

Keretfeldolgozási idő + átviteli késés + propagációs késés ( $T_f$ )

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

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

S: átjutott keretek

G: összesen elküldött keretek

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

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

S: átjutott keretek

G: összesen elküldött keretek

- 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

- Mit nevezünk **sebezhetőségi idő**nek?

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

- 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 újakezdi a procedúrát

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

- Folytonos időmodell

- Küldés előtt belehallgat

- Ha foglalt, akkor véletlen ideig vár, majd újakezd

- Ha szabad, küld

- Ütközéskor véletlen ideig vár, majd újakezdi a procedúrát

- 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

- 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.

- Mi a **Binary exponential backoff** módszer?

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

- Ennyi keretidőt várjunk az újraküldésig

- $n$  felső határa 10, 16 sikertelen próba után eldobjuk

- Hogyan működik az **alapvető bittérkép eljárás**?

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

- 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

- 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

- Hogyan működik az **adaptív fabejárési protokoll**?

Állomások bináris fában reprezentálva 0. időrásben mindenki küld

Ha ütközés, akkor mélységi bejárás, minden rés egy csomópontoz van rendelve

Ütközés esetén megnézzük a bal és a jobb csomópontot

Ha nincs ütközés, akkor a csomópont keresése befejeződik

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

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

- 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

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

Az adatkapcsolati rétegben működő eszköz, amely LAN-ok összekapcsolását végzi - lekorlátozzák az ütközési tartományok méretet

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.

- 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.

- Ismertesse a **feszítőfa protokoll** (STP) lépéseit?

i. Az egyik bridge a gyökér

ii. Minden bridge megkeresi a legrövidebb utat hozzá

iii. 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

- 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

- 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

- Mi az **adaptív forgalomirányító algoritmusok** fő jellemzője?

A topológia és a forgalom is befolyásolhatja a döntést

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

Ha  $J$  az  $I \rightarrow K$  optimális útvonalon van, akkor  $J \rightarrow K$  optimális útvonal is ugyanerre esik.

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

- 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.

- 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

- 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.

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

i. szomszédok címének felkutatása: HELLO csomag szórása, a szomszédok válaszolnak a saját címükkel



- ii. késleltetés meghatározása: ECHO csomag küldése, a másik oldalnak azonnal vissza kell küldenie - körbeérési idő fele kb a késleltetés
  - iii. infócsomag összeállítása: feladó azon., sorszám, korérték és a szomszédok listája a késleltetésekkel.
  - iv. szétküldés elárasztással. a routerek megjegyeznek minden (sorszám,forrás) párt, és csak akkor küldik tovább, ha új
  - v. Dijkstra algoritmus lefuttatása ha minden infó megérkezett
- 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.

- 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

- Mit nevezünk **broadcasting**nak (adatszórásnak)?

Egy csomag mindenhová történő egyidejű elküldése

- 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 feszfához, 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.

- Mit nevezünk **multicastingnak** (többesküldésnek)?

Egy csomag meghatározott csoporthoz történő egyidejű elküldése

Csoportkezelés is kell hozzá:

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

- Mire szolgál a **DF bit** az IPv4 fejlécében?

Ne darabold, *don't fragment flag* a routernek  
A beérkező datagramot ne darabolja fel

- 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)

- Mire szolgál az **azonosító** (azonosítás) az IPv4 fejlécében?

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

- Mire szolgál a **darabeltolás** (fragment offset) az IPv4 fejlécében?

*A darab helyét mutatja meg a datagramon belül*

- 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

- 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 bájton ábrázoljuk, leírni bájtonként decimálisan ábrázolva, a bájtokat pontokkal elválasztva szoktuk

- 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

- 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

- 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

- Mi az a **NAT doboz** és mire szolgál?

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

- Mi az az **MTU** és mire szolgál?

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

- Hogyan működik az **MTU felderítés**?

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

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

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

- Mi az **IPv6** cím és hogyan ábrázoljuk?

Az IPv4 kibővítése. 128 bites címek,  $4.8 \cdot 10^{28}$  cím / ember. Formátum: 16 bites értékek 8 csoportba sorolva (':'-tal elválasztva), minden csoport elején szereplő nulla sorozatok elhagyhatók és csupa nulla csoportok elhagyhatók, ekkor '::'.

- Mi a **localhost IPv6** esetén?

::1

- 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

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

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

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

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

- 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
- Mi lehet a hatása egy **ICMP forráslefojtás** csomagnak?

## TODO

- Mire szolgál az **ARP** és hogyan működik?

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

- 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.

- 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

- Milyen lehetőségeket támogat a **DHCP**?

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

- Mi DHCP esetén a **cím bérlet**?

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

- 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, ...

- 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.

- 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).

- Milyen routing megoldást/protokollt alkalmaz a **BGP**?

Path vector (útvonalvektor) protokollt.

- Hogyan működik az **útvonalvektor protokoll**?

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

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

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

- 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

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

Ha van közöttük a BGP routereiket összekötő vonal.

- 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.

- Mit nevez a **BGP csonka hálózatnak**?

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

- Mit nevez a **BGP többszörösen bekötött hálózatnak**?

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

- Mit nevez a **BGP tranzit hálózatnak**?

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

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

TODO

## 8. lecture

- 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.

- 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.

- 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

- 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.

- 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.

- Mit jelent az **RTO**, és hol használják?

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

- 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  $\alpha$  : 0.8–0.9 (0.875 a legtöbb TCP esetén).

- Mit mondhatunk **TCP esetén a hibadetektálásról**?

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

- 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.

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

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

- Mit nevezünk **folyamvezérlésnek**?



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

- Mit nevezünk **torlódásnak TCP esetén?**

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

Amikor a hálózat terhelése nagyobb mint a kapacitás

- Mi a **TCP Nagle** algoritmus működési alapelve?

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

- 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.

- Vázzon 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

- Mi az a **torlódási ablak**? Mire szolgál?

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

- Mi az a **slow start** TCP esetén?

Exponenciális növekedés (hisztórikus elnevezés: korábban még agresszívebb sémák)

- 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.

- 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.

- 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.

- 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.

- 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.

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

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

- Mely **TCP variánsok** használatosak napjainkban?

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

- 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.  $wnd = \min\{cwnd + dwnd, adv\}$  Hátrány: folyamatos RTT becslést igényel.

- Hogyan működik a **CUBIC TCP**?

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

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

C konstans

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

- Mik a TCP problémái kis folyamatok esetén?

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

- Mik a **TCP problémái vezeték nélküli hálózatok esetén?**

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

- 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

- Mit nevezünk **munkamenetnek** az ISO/OSI referencia modellben?

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

- Mit tud a **DNS** tartománynevek (körzetnevek) rendszeréről?

**TODO**

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

*Top Level Domain*

.com

.edu

.net

.org\

- Mik azok a **DNS erőforrás rekordok**? Mit tárolnak (1-2 példa)?

TODO

- Mit tud a (DNS) **zónáról**?

TODO

- A névfeloldásnál mit neveznek **iteratív lekérdezésnek**? Mik a jellemzői?

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

- A névfeloldásnál mit neveznek **rekurzív lekérdezésnek**? Mik a jellemzői?

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

- Írja le a **lokális névszerverek** legfőbb jellemzőit!

TODO

- Mit jelent **DNS esetén a cache**? Mire jó?

TODO

- Ismertesse egy **HTML** oldal lekérésének 5 lépését!

TODO

- Mit nevezünk **statikus weboldalnak**?

A statikus weboldal tartalma nem változik csak manuális átszerkesztéssel.

- Mit nevezünk **dinamikus weboldalnak**?

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

- Mi az a **PLT**? Mit mérünk vele?

TODO

- Mik azok a **párhuzamos és perzisztens kapcsolatok**?

TODO

- Hogyan működik a **cache HTTP esetén**?

TODO

- Mire jó egy **HTTP proxy**? Hogyan működik?

TODO

- Mi a **CDN**? Milyen problémát old meg? Hogyan valsítja ezt meg?

TODO

- Mik a **p2p** hálózatok legfontosabb jellemzői?

TODO

- Mi a szerepe egy **pér**-nek egy p2p hálózatban?

TODO

- Mik egy **torrent** letöltésének lépései (4 lépés)?

TODO

- Mit nevezünk **choke pér**-nek?

TODO

- Mi az a **séd pér**?

TODO

 AlexAegis / elte-cn

Code

Issues


Pull requests

Actions

Projects

Security

Insights

 master ▾

...

elte-cn / quiz / quiz.md



AlexAegis fixes

 2 contributors

Raw

Blame



636 lines (348 sloc) 26.8 KB

# Quiz

- Az ISO/OSI modell mely rétegéhez sorlíhatók a következő fogalmak: Optikai kábel, Wifi jel, CAT6 UTP kábel?

Fizikai réteg / Physical

- Az ISO/OSI modell mely rétege foglalja magába a közeghozzáférés vezérlését (MAC)?

Adatkapcsolati réteg / Data link

- Az ISO/OSI modell mely rétege definiálja az átvitelre szánt adatok keretekre tördelését?

Adatkapcsolati réteg / Data Link

- Az ISO/OSI modell mely rétege felel az útvonal választásért?

Hálózati réteg / Network

- Az ISO/OSI modell mely rétege felel a csomagtovábbításért?

Hálózati réteg / Network

- Az ISO/OSI modell mely rétegel felel az üzenetek adott állomáson belüli forgalom multiplexálásáért/demultiplexálásáért?

Szállítói réteg / Transport

- Az ISO/OSI modell mely rétegéhez tartozik a TCP protokoll?

Szállítói réteg / Transport

- Az ISO/OSI modell mely rétegéhez tartozik az UDP protokoll?

Szállítói réteg / Transport

- Az ISO/OSI modell mely rétege felelhet szinkronizációs pont menedzsmentért (checkpoint beszúrása, stb.)?

Munkamenet (Ülés) réteg / Session

- Az ISO/OSI modell mely rétege felel az adatkonverzióért különböző reprezentációk között?

Megjelenítési réteg / Presentation

- Az ISO/OSI modell mely rétegéhez sorolhatók a következő fogalmak: BitTorrent, HTTP, BitCoin kliens?

Alkalmazási réteg / Application

- Az ISO/OSI mely rétegeit nem használjuk az Internet architektúrájának leírásához? (Segítség: avagy mely rétegek nem képeik részét a bevezetett hibrid modellnek?)

Megjelenítési réteg / Presentation

Munkamenet (Ülés) réteg / Session

- Adott két végpont, melyeket egy switch/router és a közöttük lévő két fizikai link kapcsol össze. Mit nevezünk feldolgozási késleltetésnek (processing delay) egy csomag átvitele esetén?

Azt az időt, amit a routeren a csomag fejléceinek feldolgozása és továbbítási döntésének meghozatala igényel.

- Adott két végpont, melyeket egy switch/router és a közöttük lévő két fizikai link kapcsol össze. Mit nevezünk sorban-állási késleltetésnek (queueing delay) egy csomag átvitele esetén?

Azt az időt, amit a csomag a switch/router várakozási sorában várakozással tölt.

- Adott egy fizikai link, ami két eszközt kapcsol össze, melyek kommunikálni szeretnének. Mit nevezünk propagációs késésnek (propagation delay) ebben az esetben?

Azt az időt, ami a jelnek szükséges ahhoz, hogy áthaladjon a fizikai közegen, ami összeköti a küldő és a cél eszközöket.

- Adott két végpont, melyek között egy 120MB-os (megabájtos) fájl letöltése 2 percet vesz igénybe. Mekkora az átviteli ráta (throughput) a két oldal között? A választ Mpbs-ben (megabits/second) adja meg! (Segítség: 1 Mpbs =  $10^6$  bps, 1MB =  $10^6$  Bájtt)

8.0

- Mennyi az átviteli késleltetése egy 1500 bájtos csomagnak egy olyan hálózaton, ahol az elérhető adatráta 12 Gbps? A választ mikromásodpercben (us) adjuk meg! (segítség: 1us =  $10^{-6}$  sec, 1Gbps =  $10^9$  bps (bits/second))

1.0

- Egy optikai gerinchálózaton két routert 200km üvegszál köti össze. Az üvegszálban a jelterjedési sebesség  $2 \times 10^8$  m/s. Mekkora propagációs késést tapasztalunk a fenti optikai linken ezredmásodpercben kifejezve (ms)? (segítség: 1ms =  $10^{-3}$  s)

1.0

- Melyik állítások igazak a csomagtovábbításra (forwarding)?

A csomagot egy kimenő vonal felé irányítja

Időskála: nanosecundum

Adat síkban (data plane) valósul meg

Helyi folyamat

- Melyik állítások igazak az útvonal-meghatározásra (routing)?

A csomagok által követendő útvonalak kiszámítása

Időskála: kb. 10 ezredmásodperc

Vezérlési rétegben valósítják meg

Globális folyamat

- Melyik állítások igazak a kapcsolatállapot (link state) alapú routing protokollra?



Megméri a szomszédokhoz vezető költséget, majd ezt elküldi minden routernek.

Dijkstra algoritmust alkalmaz

- Melyik állítások igazak a távolságvektor (distance vector) alapú routing protokollra?

Aszinkron működés.

Lényegében elosztott Bellman-Ford algoritmus.

Minden router csak a szomszédjával kommunikál.

- Melyik állítások igazak az alternáló bit protokollra (ABP)?

Küldő egyesével küldi a sorszámmal ellátott kereteket (kezdetben 0-s sorszámmal) és addig nem küld újat, még nem kap nyugtát a vevőtől egy megadott határidőn belül.

Vevő oldalon, ha nincs hiba az adatrészt továbbküldi a hálózati rétegnek, végül nyugtázza

- Adott egy Distance Vector protokollt használó hálózat. Az  $u$  állomás szomszédai  $A$ ,  $B$  és  $C$  állomások. Adottak az alábbi élköltségek:  $c(u,A) = 3$ ,  $c(u,B) = 1$ ,  $c(u,C) = 7$ . Az  $u$  állomás egy adott időpillanatban megkapja mindhárom szomszéd távolságvektorait:  $dA(B) = 12$ ,  $dA(C) = 3$ ,  $dA(D) = 4$ ,  $dB(A) = 3$ ,  $dB(C) = 8$ ,  $dB(D) = 2$ ,  $dC(A) = 1$ ,  $dC(B) = 2$ ,  $dC(D) = 1$

$u$  vektorainak frissítése után adjuk meg  $dU(A)$  távolságot!

3.0

- Adott egy Distance Vector protokollt használó hálózat. Az  $u$  állomás szomszédai  $A$ ,  $B$  és  $C$  állomások. Adottak az alábbi élköltségek:  $c(u,A) = 3$ ,  $c(u,B) = 1$ ,  $c(u,C) = 7$ . Az  $u$  állomás egy adott időpillanatban megkapja mindhárom szomszéd távolságvektorait:  $dA(B) = 12$ ,  $dA(C) = 3$ ,  $dA(D) = 4$ ,  $dB(A) = 3$ ,  $dB(C) = 8$ ,  $dB(D) = 2$ ,  $dC(A) = 1$ ,  $dC(B) = 2$ ,  $dC(D) = 1$   $u$  vektorainak frissítése után adjuk meg  $dU(D)$  távolságot!

3.0

- Adott egy Distance Vector protokollt használó hálózat. Az  $u$  állomás szomszédai  $A$ ,  $B$  és  $C$  állomások. Adottak az alábbi élköltségek:  $c(u,A) = 3$ ,  $c(u,B) = 1$ ,  $c(u,C) = 7$ . Az  $u$  állomás egy adott időpillanatban megkapja mindhárom szomszéd távolságvektorait:  $dA(B) = 12$ ,  $dA(C) = 3$ ,  $dA(D) = 4$ ,  $dB(A) = 3$ ,  $dB(C) = 8$ ,  $dB(D) = 2$ ,  $dC(A) = 1$ ,  $dC(B) = 2$ ,  $dC(D) = 1$   $u$  vektorainak frissítése után adjuk meg  $dU(C)$  távolságot!

6.0

- Hogyan tanulják meg a switch-ek a forrás állomás címét?

Ha egy A porton érkezik egy csomag, melyet B állomásnak küldtek, és B nem szerepel a továbbítási táblában, akkor megtanulja, hogy B állomás az A port irányában érhető el.

- Egy globális továbbítási állapot (global forwarding state) akkor és csak akkor érvényes ha... (Több helyes válasz is lehet.)

Nincsenek hurkok/körök a hálózatban.

Nincsenek zsákutcák (dead ends) a hálózatban.

- A adat sík (data plane) a csomagok feldolgozásáért és továbbításáért felel.

Igaz.

- A adat sík (data plane) a router agya, ami pl. a konfigurálásért, az útvonalmeghatározásért és statisztikák vezetéséért felel.

Hamis.

- A vezérlési sík (control plane) a router agya, ami pl. a konfigurálásért, az útvonalmeghatározásért és statisztikák vezetéséért felel.

Igaz.

- Mi a fő probléma a forrás-cél alapú csomagtovábbítással (source- and destination-based forwarding)?

A továbbítási táblákban sokkal több ( $-n^2$ ) bejegyzést kell nyilvántartani, mint a cél-alapú megoldásnál.

- Mi igaz egy hálózat C végpontjához a készített feszítőfájára? (A hálózat routerekből és végpontokból áll. Tegyük fel, hogy a cél, amihez a feszítőfát elkészítjük csak végpont lehet.) Több válasz is helyes!

C minden routerből elérhető a feszítőfa élei mentén.

Minden router egy kimenő éllel rendelkezik.

Minden routert tartalmaz.

- Mikor érvényes egy globális továbbítási állapot (global forwarding state)?

Ha a csomagokat mindig leszállítja a célállomásnak.

- Mikor használ egy switch elárasztást egy csomag továbbítása során?

Ha a csomag célállomása nem szerepel a továbbítási táblában.

- Mely állítások igazak a fizikai rétegre?
  - Szolgáltatása, hogy információt (biteket) visz át két fizikailag összekötött eszköz között
- Mely állítások igazak a Link-State Routing-re?
  - A hálózat globális szerkezetét (topológiáját) igényli.
  - Lokálisan minden router egy Dijkstra algoritmust futtat.
  - Elárasztással, minden routernek eljuttatja a lokális információkat.
- Mely állítások igazak a végpont-végpont megbízhatóságra?
  - A végpont-végpont megbízhatóságot az L4 (Transport - Szállítói) réteg biztosítja.
  - A hálózat legyen a lehető legegyszerűbb, azaz nem biztosít végpont-végpont megbízhatóságot.
  - Az alkalmazásoknak nem kell a hálózati problémákkal foglalkozniuk, így a megbízhatóság biztosításával sem.
- Mely állítások igazak az alapsávú átvitelre?
  - a digitális jel direkt árammá vagy feszültséggé alakul
  - a jel minden frekvencián átvitelre kerül
- Mely állítások igazak az szélessávú átvitelre?
  - egy széles frekvencia tartományban történik az átvitel, nem minden frekvencián kerül átvitelre a jel **NOT VERIFIED**
  - a jelet modulálással ülteti egy vivóhullámra **NOT VERIFIED**
- Mely állítások igazak a Hamming-kódra? (3 állítás igaz)
  - Mindegyik ellenőrző bit a bitek valamilyen csoportjának a paritását állítja be párosra (vagy páratlanra)
  - 2 egészhatvány sorszámú pozíciói lesznek az ellenőrző bitek, azaz 1,2,4,8,16,..., a maradék helyeket az üzenet biteivel töltjük fel
  - Paritást használó technika
- Mely állítások igazak a csúszóablak protokollra?

Csak duplex csatorna esetén alkalmazható. Adat és nyugta csomagok egyszerre utazhatnak.

A keret nyugtázója tartalmazza a következőnek várt keret sorozatszámát.

A nem megengedett sorozatszámmal érkező kereteket el kell dobni.

- Mely állítások igazak a simplex megáll és vár protokollra (zajos csat.)?

Csomagvesztés esetén az időzítő lejártá után (timeout) újraküldi a keretet.

Nyugta elvesztése esetén duplikátumok adódhatnak át a felsőbb rétegnek a fogadó oldalon.

- Mely állítás igaz a bridge-eknél (hidaknál) látott feszítőfa protokollra? (STP)?

Egy bridge a szomszéd bridge-eknek küldi el a konfigurációs üzenetét, mely alapján azok frissítik a gyökér csomópont és a hozzá vezető úthoz kapcsolódó információkat.

- Mely állítás igaz a bridge-eknél (hidaknál) látott feszítőfa protokollra? (STP)?

A fa gyökere a legkisebb ID-val rendelkező bridge, melyet a szomszédoktól kapott üzenetek alapján frissít egy bridge.

- Egy kód Hamming-távolsága 2. Hány egyszerű bithibát tudunk javítani ezzel a kóddal?

1

- Egy kód Hamming-távolsága 5. Hány egyszerű bithibát tudunk javítani ezzel a kóddal?

2

- Egy kód Hamming-távolsága 8. Hány egyszerű bithibát tudunk felismerni ezzel a kóddal?

7

- Egy kód Hamming-távolsága 13. Hány egyszerű bithibát tudunk javítani ezzel a kóddal?

6

- Egy kód Hamming-távolsága 25. Hány egyszerű bithibát tudunk javítani ezzel a kóddal?

12

- Egy kód Hamming-távolsága 15. Hány egyszerű bithibát tudunk felismerni ezzel a kóddal?

14

- A megbízható adatátvitel 4 fő célja közül melyik szól az adat leszállítási idejének minimalizálásáról.

Időbeliség/Timeliness

- A megbízható adatátvitel 4 fő célja körül melyik szól arról, hogy:

"az adat leszállítása biztosított, sorrend helyes és átvitel során nem módosul".

Helyesség/Correctness

- Egy csúszóablak (sliding window) protokoll esetén a sorszámok tere 0,1,2,3,4,5,6,7, a négy hosszú küldési ablakban az 1,2,3,4 sorszámok vannak. Az 1-es sorszámú nyugta beérkezése után, milyen sorszámmal lehetnek elküldött de nem nyugtázott csomagok.

2,3,4,5

- Egy csúszóablak (sliding window) protokoll esetén a sorszámok tere 0,1,2,3,4,5,6,7. A fogadó 2 csomagot tud pufferelni, a vételi ablakában 2,3 sorszámok szerepelnek. Mit tesz a fogadó egy 1-es sorszámú csomag beérkezése esetén?

Eldobja a csomagot és nyugtát küld.

- Melyik nyugtázási módszerre igaz az alábbi állítás? A nyugta a legnagyobb sorszámot tartalmazza, amelyre igaz, hogy az összes kisebb (vagy egyenlő) sorszámú csomag már sikeresen megérkezett a vevőhöz.

Kumulatív nyugta - cumulative ACK

- Melyik nyugtázási módszerre igaz az alábbi állítás? Teljes információt ad a forrásnak és jól kezeli a nyugták elvesztését is, azonban az a nagy hálózati overheadje miatt csökkenti a teljesítményt.

Teljes információ visszacsatolás - Full Information Feedback

- Kumulatív nyugta (cumulative ACK) esetén miként tudjuk detektálni a csomagvesztést?

Az izolált csomagvesztéseket nyugta duplikátumok jelzik. Emelett timerekkel is dolgozik a módszer.

- Hogyan definiáltuk a helyességet! Egy szállítási mechanizmus helyes, akkor és csak akkor...

Minden elvesztett vagy hibás csomagot újraküld.

- Adott egy hálózat:

A-----1 Gbps-----B-----10 Gbps-----C

és adott 3 folyam:

- i. folyam: A-ból B-be küld adatot
- ii. folyam: B-ból C-be küld adatot
- iii. folyam: A-ból C-be küld adatot

Milyen rátát kap a 2. folyam Mbps-ben kifejezve, ha max-min fair allocation-t alkalmazunk a sávszélességek kiosztására (a fenti példában)?

9500.0 (megközelítőleg : 0.0)

9.5 (megközelítőleg: 0.0)

- Mi a folyam vezérlés (flow control) célja a megbízható adatátvitel során?

A lassú vevő túlterhelésének megakadályozása.

- Mik történhetnek egy csomaggal átvitel során, melyet egy megbízható végpont-végpont adattranszport protokollnak kezelnie kell?

csomagvesztés - loss

meghibásodás - being corrupted

duplikátumok - duplicates

várakoztatás - being delayed

csomagok sorrendjének megváltoztatása - reordering

- Jelölje be, hogy az állítások mely multiplexálási technikákra igazak!

- A teljes frekvencia tartományt szűkebb sávokra bontja

**Frekvencia multiplexálás**

- Vezetékes kommunikáció esetén minden egyes csatornához külön pont-pont fizikai kapcsolat tartozik

**Térbeli multiplexálás**

- Vezeték nélküli kommunikáció esetén minden egyes csatornához külön antenna rendelődik

## Térbeli multiplexálás

- Minden állomás saját frekvencia tartományt kap

## Frekvencia multiplexálás

- Diszkrét időszelvények használata

## Idő-osztásos multiplexálás (TDM)

- Minden állomás saját időszelvényt kap

## Idő-osztásos multiplexálás (TDM)

- Mit nevezünk elnyelődésnek?

A küldési és vételi energiák hányadosát.

- Egy  $s(t)$  függvényt a  $\sin(t)$  vivőhullámra a következőképp kódolunk:  $s(t) \cdot \sin(t)$   
Melyik modulációs technikát alkalmaztuk?

Amplitúdó moduláció

- Egy  $s(t)$  függvényt a  $\sin(t)$  vivőhullámra a következőképp kódolunk:  $\sin(t) \cdot s(t)$   
Melyik modulációs technikát alkalmaztuk?

Frekvencia moduláció

- Egy  $s(t)$  függvényt a  $\sin(t)$  vivőhullámra a következőképp kódolunk:  $\sin(t + s(t))$   
Melyik modulációs technikát alkalmaztuk?

Fázis moduláció

- Mely modulációs technika használja a vivőhullám több jellemzőjét is a szimbólumok kifejezésére?

QAM-16 technika

- A 100 Mbps Ethernetnél alkalmazott 4/5 kódolással \_ %-ot veszítünk a hatékonyságból!

20

- Két szimbólum használata esetén a szimbólum ráta 4 Baud. Négy szimbólum használata mellett mekkora lesz a szimbólum ráta, ha semmi mást nem változtatunk?

4 Baud

- Négy szimbólum használata esetén hány bitet tudunk egy szimbólumba kódolni?

2

- Mi az összefüggés a frekvencia (f), a hullámhossz (L (LAMBDA)) és a fénysebesség (c) között?

$$f \cdot L = c$$

- Mekkora következő két bitsorozat Hamming-távolsága? d( 1001, 1011 )

1

- Mekkora következő két bitsorozat Hamming-távolsága? d( 11111, 01011 )

2

- Mekkora következő két bitsorozat Hamming-távolsága? d( 11111, 11000 )

3

- Minek kell teljesülnie a chip vektorokra a CDMA módszer esetén?

Páronként ortogonális vektoroknak kell lenniük.

- Adott három állomás (A,B,C), melyek CDMA módszert használnak. Jelölje be, hogy mely chip vektorok lennének helyesek?

A: (1,0,0) ; B: (0,1,0) ; C: (0,0,1)

A: (1,1,0) ; B: (1,-1,0) ; C: (0,0,-1)

- Alkosson párokat a keretezési technikák jellemzőiből és neveiből!
  - A fogadó az adatban előforduló minden 11111 részsorozat után ellenőrzi a következő bitet, majd ez alapján lép tovább.

#### Bit beszúrás

- Nagyon érzékeny a bithibákra (pl. fejléc meghibásodása)

#### Karakterszámlálás

- Egy speciális ESC (Escape) bájtot szúr be az "adat" ESC bájtok elé

#### Bájt beszórás

- SONET hálózatoknál alkalmazzák

#### Óra alapú keretezés

- Mely szolgáltatásokért felel az adatkapcsolati réteg? (4 állítás helyes)



Per-hop megbízhatóság

Per-hop hibakezelés

Adatok keretekre tördelése

Közeghozzáférés

- Az alábbi három kódolás közül melyiket érdemes használni, ha tudjuk, hogy a csatorna nem megbízható.  $R(S)$  jelöli a kód rátáját,  $q(S)$  pedig a kód távolságát!

$R(S) = 0.7$  és  $q(S) = 0.7$

- Legyen  $d(x,y)$  két kódszó Hamming-távolsága. Hogyan definiálja egy  $S$  kód Hamming-távolságát?

Az  $S$ -beli kódszó párok Hamming távolságainak a minimuma.

- Egy protokoll CRC-t használ hiba felismeréséhez. Az alkalmazott generátor polinom fokszáma 4. Hány biten ábrázolható a CRC kontrollösszeg (a maradék polinom)?

4

- Egy protokoll CRC-t használ hiba felismeréséhez. Az alkalmazott generátor polinom fokszáma 7. Hány biten ábrázolható a CRC kontrollösszeg (a maradék polinom)?

7

- Egy protokoll CRC-t használ hiba felismeréshez. Az alkalmazott generátor polinom fokszáma 10. Hány biten ábrázolható a CRC kontrollösszeg (A maradék polinom)?

10

- Egy protokoll CRC-t használ hiba felismeréshez. Az alkalmazott generátor polinom fokszáma 12. Hány biten ábrázolható a CRC kontrollösszeg (A maradék polinom)?

12

- Egy protokoll CRC-t használ hiba felismeréséhez. Az alkalmazott generátor polinom fokszáma 32. Hány biten ábrázolható a CRC kontrollösszeg (a maradék polinom)?

32

- Az előadáson látott naiv hibadetektáló megoldás minden keretet kétszer küld el. Ezt követően a két kópia egyezését a hibamentes átvitel eldöntésére. Mely állítások igazak erre a módszerre? (2 állítás igaz)

Túl nagy a költsége.

Gyenge hibavédelemmel rendelkezik.

- Mely bithibát nem képes felismerni a CRC módszer, ha a generátor polinom  $x^3 + x + 1$ , ahol  $x^4$  jelöli az "x a negyediken" hatványt?

ahol a hiba polinom  $E(x) = x^4 + x^2 + x$

- Mely bithibát nem képes felismerni a CRC módszer, ha a generátor polinom  $x^4 + x + 1$ , ahol  $x^4$  jelöli az "x a negyediken" hatványt?

ahol a hiba polinom  $E(x) = x^5 + x^2 + x$

- Mely bithibát nem képes felismerni a CRC módszer, ha a generátor polinom  $x^9 + x^2 + x + 1$ , ahol  $x^4$  jelöli az "x a negyediken" hatványt?

ahol a hiba polinom  $E(x) = x^{11} + x^4 + x^3 + x^2$

- Mely bithibát nem képes felismerni a CRC módszer, ha a generátor polinom  $x^{11} + x^9 + x + 1$ , ahol  $x^4$  jelöli az "x a negyediken" hatványt?

ahol a hiba polinom  $E(x) = x^{12} + x^{10} + x^2 + x$

- Mely bithibát nem képes felismerni a CRC módszer, ha a generátor polinom  $x^{32} + x^{31} + x + 1$ , ahol  $x^4$  jelöli az "x a negyediken" hatványt?

ahol a hiba polinom  $E(x) = x^{33} + x^{32} + x^2 + x$

- Mely csatornára igaz az alábbi állítás? A kommunikáció pusztán az egyik irányba lehetséges

Szimplex csatorna

- Mely csatornára igaz az alábbi állítás? Mindkét irányba folyhat kommunikáció, de egyszerre csak egy irány lehet aktív.

Fél-duplex csatorna

- Mely csatornára igaz az alábbi állítás? Mindkét irányba folyhat a kommunikáció szimultán módon.

Duplex csatorna

- Adott N állomás, melyek Alapvető bittérkép protokollt használnak a közeghozzáféréshez. A versengési időrés 1 időegység. Egy adatkeret küldése szintén egységesen 1 időegységig tart. Legrosszabb esetben hány időegységet kell egy állomásnak várnia a saját keretre átvitelének megkezdése előtt? [Azt az időrést már ne számoljuk, amiben a saját keret is átvitelre kerül. Továbbá tegyük fel, hogy kötetlenül a versengési időrés előtt állunk]!

N

- Adott N állomás, melyet bináris visszaszámlálás protokollt (Mok és Ward féle javítás nélkül) használnak a közeghozzáféréshez. A versengési időrés 1 időegység. Egy adatkeret küldése szintén egységesen 1 időegységig tart. Legrosszabb esetben hány időegységet kell állomásnak várnia a saját kertre átvitelének megkezdése előtt? [Azt az időrést már ne számoljuk, amiben a saját keret is átvitelre kerül. Tovább tegyük fel, hogy közvetlenül a versengési időrés előtt állunk.]

Soha nem kerül átvitelre az állomás kerete.

- Adott N állomás, melyet bináris visszaszámlálás protokollt (Mok és Ward féle javítás nélkül) használnak a közeghozzáféréshez. A versengési időrés 1 időegység. Egy adatkeret küldése szintén egységesen 1 időegységig tart. Legrosszabb esetben hány időegységet kell állomásnak várnia a saját kertre átvitelének megkezdése előtt? [Azt az időrést már ne számoljuk, amiben a saját keret is átvitelre kerül. Tovább tegyük fel, hogy közvetlenül a versengési időrés előtt állunk.]

1

- Adott N állomás, melyek bináris visszaszámlálás protokollt és Mok és Ward féle javítást használnak a közeghozzáféréshez. 4 állomás áll készen keret küldésére. A versengési időrés 1 időegység. Egy adatkeret küldése szintén egységesen 1 időegységig tart. Legrosszabb esetben hány időegységet kell egy állomásnak várnia a saját kerete átvitelének megkezdése előtt? [Azt az időrést már ne számoljuk, amiben a saját keret átvitelre kerül. Továbbá tegyük fel, hogy közvetlenül a versengési időrés előtt állunk! Nem érkeznek újabb kérések a rendszerbe!]

7

- Adott N állomás, melyek bináris visszaszámlálás protokollt és Mok és Ward féle javítást használnak a közeghozzáféréshez. 4 állomás áll készen keret küldésére. A versengési időrés 1 időegység. Egy adatkeret küldése szintén egységesen 1 időegységig tart. Legrosszabb esetben hány időegységet kell egy állomásnak várnia a saját kerete átvitelének megkezdése előtt? [Azt az időrést már ne számoljuk, amiben a saját keret átvitelre kerül. Továbbá tegyük fel, hogy közvetlenül a versengési időrés előtt állunk!]

1

- Mi igaz a bridge-eknél (hídnál) látott MAC címek tanulása módszerre?

A beérkező keretben szereplő forrásállomás MAC címét és a beérkezési portot betesszük a továbbítási táblába.

- Mit jelent az optimalitási elv útvonalkiválasztás esetén?

Legyen P az I-ből K állomásba vezető optimális útvonal. Ekkor bármely J állomást véve a P útvonal mentén, a J-ből K-ba vezető optimális útvonal P-re esik (annak része).

- Egy távolságvektor routing protokollt használó hálózatban az A állomás routing táblája a következő:

host	költség	next hop
B	7	B
C	10	C
D	1	D
E	14	D

B szomszédától a következő távolságvekort kapja:

C	2
D	3
E	3

Mi lesz D költsége A állomás routing táblájában?

1

- Egy távolságvektor routing protokollt használó hálózatban az A állomás routing táblája a következő:

host	költség	next hop
B	7	B
C	10	C
D	1	D
E	14	D

B szomszédától a következő távolságvekort kapja:

C	2
D	3
E	3

Mi lesz C költsége A állomás routing táblájában?

9

- Melyik állítás igaz?

Az Alternáló Bit Protokoll csatorna kihasználtsága azonos a szimplex megáll és vár protokoll esetén látottal.

- Melyik állítás igaz?

Switchek esetén nincs szükség CSMA/CD-re.

Switchek esetén full duplex linkek kötik be az állomásokat.

- Melyik állítás igaz?

A pipelineing technika nem segít a csatornakihasználtság javításában.

- Melyik állítás igaz?

Minden switch egyben bridge is.

- Adott  $2^N$  (kettő az N-ediken) állomás, melyek adaptív fabejárás protokollt használnak a közeghozzáféréshez. 2 állomás áll készen keret küldésére, melyek ütközést okoznak. Egy adatkeret küldése egységesen 1 időegységig tart. Legrosszabb esetben hány időegység szükséges az ütközés feloldásához? (Az első ütközést okozó időrést nem számolva hány időegység alatt fut le az ütközés feoldás. Továbbá tegyük fel, hogy nem érkeznek újabb kérések a rendszerbe!)

2N

- Adott  $2^N$  (kettő az N-ediken) állomás, melyek adaptív fabejárás protokollt használnak a közeghozzáféréshez. 2 állomás áll készen keret küldésére, melyek ütközést okoznak. Egy adatkeret küldése egységesen 1 időegységig tart. Legjobb esetben hány időegység szükséges az ütközés feloldásához? [Az első ütközést okozó időrést nem számolva hány időegység alatt fut le az ütközés feloldás. Továbbá tegyük fel, hogy nem érkeznek újabb kérések a rendszerbe!]

2

- Adott 8 állomás, melyek adaptív fabejárás protokollt használnak a közeghozzáféréshez. Az állomások sorszámai 1-8, melyek a fa levél szintjén helyezkednek el balról jobbra. A 3. és 4. állomások akarnak keretet átvinni a csatornán. A lent látható időrés sorozatok közül melyik tartozik a fenti ütközés feloldásához? [Az első ütközést okozó időrést ne vegyük figyelembe. Továbbá tegyük fel, hogy nem érkeznek újabb kérések a rendszerbe!]

ütközés (3 és 4 küld) | üres | ütközés (3 és 4 küld) | 3 küld | 4 küld | üres



szarkadani No commit message



2 contributors



Raw

Blame



326 lines (247 sloc) 16.1 KB

# Algorithms

## CRC

$G(x)$  egy generátor polinom, aminek a foka  $r$ , ezt a polinomot a küldő és a vevő egyaránt ismeri.

## Algoritmus

1. Fűzzünk  $r$  darab 0 bitet a keret alacsony helyiértékű végéhez, tehát az  $m + r$  bitet fog tartalmazni és reprezentálja a  $x^r * M(x)$  polinomot
2.  $x^r * M(x)$  elosztása  $G(x)$ -szel modulo 2 (bitsorozatok)
3. Az előző osztás maradékának kivonása  $x^r * M(x)$ -ből modulo 2 (bitsorozatok), az eredmény az ellenőrző összeggel ellátott, továbbítandó keret. Jelölése:  $T(x)$
4. A vevő a  $T(x) + E(x)$  polinomnak megfelelő sorozatot kapja, ahol  $E(x)$  a hibapolinom. Ezt elosztja a  $G(x)$  generátor polinommal, ha van maradéka ennek az osztásnak, akkor hiba történt.

## CDMA

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 tulajdonságát a multiplexáláshoz.

## Algoritmus

1. Minden bitidőt  $m$  darab rövid intervallumra osztunk, ezek a töredékek (angolul chip)
2. Minden állomáshoz egy  $m$  bites kód tartozik, úgynevezett töredéksorozat (angolul chip sequence)
3. Ha 1-es bitet akar továbbítani egy állomás, akkor elküldi a saját töredéksorozatát
4. Ha 0-es bitet akar továbbítani egy állomás, akkor elküldi a saját töredéksorozatának egyes komplementjét
5.  $m$ -szeres sávszélesség válik szükségessé, azaz szórt spektrumú kommunikációt valósít meg
6. Szemléltetésre bipoláris kódolást használunk:

Bináris 0 esetén  $-1$ , bináris 1 esetén  $+1$   
az állomásokhoz rendelt töredék sorozatok páronként ortogonálisak

## CSMA

### Az 1-perzisztens CSMA protokoll:

- Működése:
  - Vivőjelérzékelés van, azaz minden állomás belehallgathat a csatornába.
  - Folytonos időmodellt használ a protokoll.
  - Keret leadása előtt belehallgat a csatornába:
    - a. Ha foglalt, akkor addig vár, amíg fel nem szabadul. Szabad csatorna esetén azonnal küld. (perzisztens)
    - b. Ha szabad, akkor küld.
  - Ha ütközés történik, akkor az állomás véletlen hosszú ideig vár, majd újratekdi a keret leadását.
- Tulajdonságok:
  - A terjedési késleltetés nagymértékben befolyásolhatja a teljesítményt.
  - Jobb teljesítményt mutat, mint az ALOHA protokollok.



## A nem-perzisztens CSMA protokoll:

- Működése:
  - Vivőjelérzékelés van, azaz minden állomás belehallgathat a csatornába.
  - Folytonos időmodellt használ a protokoll.
  - Mohóság kerülése.
  - Keret leadása előtt belehallgat a csatornába:
    - a. Ha foglalt, akkor véletlen ideig vár (nem figyeli a forgalmat), majd kezd előről a küldési algoritmust. (nem-perzisztens)
    - b. Ha szabad, akkor küld.
  - Ha ütközés történik, akkor az állomás véletlen hosszú ideig vár, majd újakezdi a keret leadását.
- Tulajdonságok:
  - Jobb teljesítményt mutat, mint az 1-perzisztens CSMA protokoll. (intuitív)

## A p-perzisztens CSMA protokoll:

- Működése:
  - Vivőjel érzékelés van, azaz minden állomás belehallgathat a csatornába.
  - Diszkrét időmodellt használ a protokoll.
  - Adás kész állapotban az állomás belehallgat a csatornába:
    - a. Ha foglalt, akkor vár a következő időrésig, majd megismétli az algoritmust.
    - b. Ha szabad, akkor  $p$  valószínűséggel küld, illetve  $1-p$  valószínűséggel visszalép a szándékától a következő időrésig. Várakozás esetén a következő időrésben megismétli az algoritmust. Ez addig folytatódik, amíg el nem küldi a keretet, vagy amíg egy másik állomás el nem kezd küldeni, mert ilyenkor úgy viselkedik, mintha ütközés történt volna.
  - Ha ütközés történik, akkor az állomás véletlen hosszú ideig vár, majd újakezdi a keret leadását.

## A CSMA/CD protokoll:

- (CD → Collision Detection: ütközés érzékelés) Ütközés érzékelés esetén meg lehessen szakítani az adást. („Collision Detection”):
- Működése:
  - 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 újra elkezdi leadni a keretét.
- Az ütközés detektálás minimális ideje az az idő, ami egy jelnek a két legtávolabbi állomás közötti átviteléhez szükséges.
- Egy állomás megszerezte a csatornát, ha minden más állomás érzékeli az átvitelét.
- Az ütközés detektálás működéséhez szükséges a keretek hosszára egy alsó korlátot adnunk
- Ethernet a CSMA/CD-t használja
- Alapvetés: a közeg lehetőséget ad a csatornába hallgatásra
- Gyér forgalom esetén a közeghozzáférés nagyon gyors, mivel kevés állomás kíván a csatornán adni. Nagy hálózati forgalom esetén az átvitel lelassul, mivel a nagy csatorna terhelés miatt gyakoriak lesznek az ütközések. (A széles körben elterjedt Ethernet hálózat ezt a módszert használja.)

## Algoritmus

1. Használjuk valamely CSMA variánst
2. A keret kiküldése után, figyeljük a közeget, hogy történik-e ütközés
3. Ha nem volt ütközés, akkor a keretet leszállítottuk
4. Ha ütközés történt, akkor azonnal megszakítjuk a küldést. Miért is folytatnánk hisz a keret már sérült...
5. Alkalmazzuk az bináris exponenciális hátralék módszert az újraküldés során (binary exponential backoff)

Ütközések történhetnek, az ütközéseket gyorsan észleljük és felfüggesztjük az átvitelt.

## ALOHA:

### Egyszerű ALOHA protokoll

A csatornakiosztás problémáját oldja meg. A rendszer lényege hogy a felhasználó bármikor adhat, ha van továbbítandó adata. De ha bárki bármikor adhat, akkor valószínű, hogy ütközések lesznek. A küldő azonban figyelheti a csatornát, így meg tudja állapítani hogy a keret tönkrement-e vagy sem. Ütközés esetén véletlen ideig vár az újraküldéssel.

Tulajdonságok:

- ALOHA protokollok áteresztő képessége egyforma keretméret esetén maximális.

- Keret idő – egy szabványos, fix hosszúságú keret átviteléhez szükséges idő
- Tegyük fel, hogy a felhasználók végtelen populációja a kereteket Poisson-eloszlás szerint állítja elő.
- Keretidőnként átlagosan  $N$ -et, ha:
  - $N > 1$ , akkor a csatorna túlterhelt.
  - $0 < N \leq 1$ , akkor a csatorna áteresztő képessége elfogadható.
  - Tegyük még fel, hogy keretidőnként  $k$  számú új és régi keret együttes elküldési kísérleteinek valószínűsége
- ugyancsak Poisson-eloszlású, és keretidőnkénti középértéke  $G$ , ha
  - $G = N$ , akkor a terhelés kicsi.
  - $G > N$ , akkor a terhelés nagy.
  - Áteresztő képesség:  $S = GP_0$ , ahol  $P_0$  keret sérülésmentes átvitelének valószínűsége.

## Réselt ALOHA protokoll

Az idő diszkrét, keretidőhöz igazodó időszegletekre osztásával az ALOHA rendszer kapacitása megduplázható. (1972, Roberts) Következmény:

- kritikus szakasz hossza a felére csökken, azaz:  $P_0 = e^{(-G)}$
- az áteresztő képesség:  $S = GP_0 = Ge^{(-G)}$
- A csatorna terhelésének kis növekedése is drasztikusan csökkentheti a médium teljesítményét.

## Távolságvektor alapú forgalomirányítás:

Minden router-nek egy táblázatot kell karbantartania, amelyben minden célhoz szerepel a legrövidebb ismert távolság, s annak a vonalnak az azonosítója, amelyiken a célhoz lehet eljutni. A táblázatokat a szomszédoktól származó információk alapján frissítik.

- Elosztott Bellman-Ford forgalomirányítási algoritmusként is nevezik.
- ARPANET eredeti forgalomirányító algoritmus ez volt. RIP (Routing Information Protocol) néven is ezt használták.

Távolságvektor alapú forgalomirányítás, Elosztott Bellman-Ford algoritmus

## Környezet és Működés:

- Minden csomópont csak a közvetlen szomszédjaival kommunikálhat.
- Aszinkron működés.

- Minden állomásnak van saját távolság vektora. Ezt periodikusan elküldi a direkt szomszédoknak.
- A kapott távolság vektorok alapján minden csomópont új táblázatot állít elő.

## Végtelenig számolás problémája:

- A „jó hír” gyorsan terjed.
- A „rossz hír” lassan terjed.
- Azaz ciklusok keletkezhetnek.
- Lehetséges megoldás:
  - „split horizon with poisoned reverse”: negatív információt küld vissza arról a szomszédjának, amit tőle „tanult”. (RFC 1058)

## Link-state

---

### Algoritmus

1. Szomszédok felkutatása, és hálózati címeik meghatározása
2. Megmérni a késleltetést vagy költséget minden szomszédhoz
3. Egy csomag összeállítása a megismert információkból
4. Csomag elküldése az összes többi router-nek
5. Kiszámítani a legrövidebb utat az összes többi router-hez (Dijkstra algoritmusát használják).

## Address Resolution Protocol (ARP)

---

### Feladata:

- Az IP cím megfeleltetése egy fizikai címnek.

### Hozzárendelés:

- Adatszóró csomag kiküldése az Ethernetre „Ki-é a 192.60.34.12-es IP-cím?” kérdéssel az alhálózaton, és mindenegyes hoszt ellenőrzi, hogy övé-e a kérdéses IP-cím. Ha egyezik az IP a hoszt saját IP-jével, akkor a saját Ethernet címével válaszol. Erre szolgál az ARP.
- Opcionális javítási lehetőségek:
  - a fizikai cím IP hozzárendelések tárolása (cache használata);
  - Leképezések megváltoztathatósága (időhatály bevezetése);
- Mi történik távoli hálózaton lévő hoszt esetén?
  - A router is válaszoljon az ARP-re a hoszt alhálózatán. (proxy ARP)

- Alapértelmezett Ethernet-cím használata az összes távoli forgalomhoz

## Bitbeszúrás

---

- Minden keret speciális bitmintával kezdődik és végződik (hasonlóan a bájt beszúráshoz)
  - A kezdő és záró bitsorozat ugyanaz
  - Például: 01111110 a High-level Data Link Protocol (HDLC) esetén
- A Küldő az adatban előforduló minden 11111 részsorozat elé 0 bitet szúr be
  - Ezt nevezzük bit beszúrásnak
- A Fogadó miután az 11111 részsorozattal találkozik a fogadott adatban:
  - 111110 -> eltávolítja a 0-t (mivel ez a beszúrás eredménye volt)
  - 111111 -> ekkor még egy bitet olvas
    - 1111110 -> keret vége
    - 1111111 -> ez hiba, hisz ilyen nem állhat elő a küldő oldalon. Eldobjuk a keretet!
- Hátránya: legrosszabb esetben 20% teljesítmény csökkenés

## Rekurzív és iteratív domainnév keresése

---

- A lekérdezésnek két fajtája van:
  - Rekurzív lekérdezés -> Ha a névszerver végzi el a névfeloldást, és tér vissza a válasszal.
  - Iteratív lekérdezés -> Ha a névszerver adja vissza a választ vagy legalább azt, hogy kitől kapható meg a következő válasz.
- Melyik a jobb?
  - Rekurzív jellemzői
    - Lehetővé teszi a szervernek a kliens terhelés kihelyezését a kezelhetőségért.
    - Lehetővé teszi a szervernek, hogy a kliensek egy csoportja felett végezzen cachelést, a jobb teljesítményért.
  - Iteratív jellemzői
    - Válasz után nem kell semmit tenni a kéréssel a névszervernek.
    - Könnyű magas terhelésű szervert építeni.

## Rekurzív DNS lekérdezés:

- A lokális szerver terhet rak a kérdezett névszerverre (pl.root)
- Honnan tudja a kérdezett, hogy kinek továbbítsa a választ?
  - Random ID a DNS lekérdezésben

## Iteratív DNS lekérdezés:

- A szerver mindig a következő kérdezendő névszerver adataival tér vissza
  - "I don't know this name, but this other server might"
- Napjainkban iteratív módon működik a DNS!!!

## Feszítőfa

---

### Algoritmus

1. Az egyik bridge-et megválasztjuk a fa gyökerének
  2. Minden bridge megkeresi a legrövidebb utat a gyökérhez
  3. Ezen utak unióját véve megkapjuk a feszítőfát
- A fa építése során a bridge-ek egymás között konfigurációs üzeneteket (Configuration Bridge Protocol Data Units [BPDUs]) cserélnek
    - A gyökér elem megválasztásához
    - A legrövidebb utak meghatározásához
    - A gyökérhez legközelebbi szomszéd (next hop) állomás és a hozzá tartozó port azonosításához
    - A feszítőfához tartozó portok kiválasztása
  - Kezdetben minden állomás feltételezi magáról, hogy gyökér - Bridge-ek minden irányba szétküldik a BPDU üzeneteiket:
    - | Bridge ID | Gyökér ID | Út költség a gyökérhez |
  - A fogadott BPDU üzenet alapján, minden switch választ:
    - Egy új gyökér elemet (legkisebb ismert Gyökér ID alapján)
    - Egy új gyökér portot (melyik interfész megy a gyökér irányába)
    - Egy új kijelölt bridge-et (a következő állomás a gyökérhez vezető úton)

## TCP

---

### Lassú indulás - Slow Start

- Cél, hogy gyorsan elérjük a könyök pontot
- Egy kapcsolat kezdetén (vagy újraindításakor)
  - $cwnd = 1$
  - $ssthresh = adv\_wnd$
  - Minden nyugtázott szegmensre:  $cwnd++$
- Egészen addig amíg
  - El nem érjük az  $ssthresh$  értéket
  - Vagy csomagvesztés nem történik

- A Slow Start valójában nem lassú
  - cwnd exponenciálisan nő

## Számos TCP változat:

- Tahoe: (az eredeti)
  - Slow start és AIMD
  - Dinamikus RTO, RTT becsléssel
- Reno:
  - fast retransmit (3 dupACKs)
  - fast recovery (cwnd = cwnd/2 vesztés esetén)
- NewReno: javított gyors újraküldés
  - Minden egyes duplikált ACK újraküldést vált ki
  - Probléma: >3 hibás sorrendben fogadott csomag is újraküldést okoz (hibásan!)
- Vegas: késleltetés alapú torlódás elkerülés

## TCP jellemzői:

- Kapcsolatorientált:
  - Két résztvevő, ahol egy résztvevőt egy IP-cím és egy port azonosít.
  - A kapcsolat egyértelműen azonosított a résztvevő párral.
  - Nincs se multi-, se broadcast üzenetküldés.
  - A kapcsolatot fel kell építeni és le kell bontani.
  - Egy kapcsolat a lezárásáig aktív.
- Megbízható:
  - Minden csomag megérkezése nyugtázásra kerül.
  - A nem nyugtázott adatcsomagokat újraküldik.
  - A fejléchez és a csomaghoz ellenőrzőösszeg van rendelve.
  - A csomagokat számozza, és a fogadónál sorba rendezésre kerülnek a csomagok a sorszámaik alapján.
  - Duplikátumokat törli.
- Kétirányú Bájtfolyam:
  - Az adatok két egymással ellentétes irányú bájtsorozatként kerülnek átvitelre.
  - A tartalom nem interpretálódik.
  - Az adatcsomagok időbeli viselkedése megváltozhat: átvitel sebessége növekedhet, csökkenhet, más késés, más sorrendben is megérkezhetnek.
  - Megpróbálja az adatcsomagokat időben egymáshoz közel kiszállítani.
  - Megpróbálja az átviteli közeget hatékonyan használni.

## Kapcsolat felépítés:

- Miért van szükség kapcsolat felépítésre?
  - Állapot kialakítása mindkét végponton
  - Legfontosabb állapot: sorszámok/sequence numbers
  - Az elküldött bájtok számának nyilvántartása
  - Véletlenszerű kezdeti érték
- Fontos TCP flag-ek/jelölő bitek (1 bites)
  - SYN – szinkronizációs, kapcsolat felépítéshez
  - ACK – fogadott adat nyugtázása
  - FIN – vége, kapcsolat lezárásához

## Kapcsolat felépítés problémája:

- Kapcsolódási zűrzavar
  - Azonos hoszt kapcsolatainak egyértelműsítése
  - Véletlenszerű sorszámmal - biztonság
- Forrás hamisítás
  - Kevin Mitnick
  - Jó random szám generátor kell hozzá!
- Kapcsolat állapotának kezelése
  - Minden SYN állapotot foglal a szerveren
  - SYN flood = denial of service (DoS) támadás
  - Megoldás: SYN cookies

## Kapcsolat lezárása:

- Mindkét oldal kezdeményezheti a kapcsolat bontását
- A másik oldal még folytathatja a küldést
  - Félig nyitott kapcsolat
  - shutdown()
- Az utolsó FIN nyugtázása
  - Sorszám + 1