1. A TCP kis folyamok (pl. HTTP üzenetek) átvitele esetén nem hatékony, hiszen a kapcsolat felépítése/lebontása sok időt vesz el. Miért nem UDP felett valósítjuk meg ezeket a szolgáltatásokat (ahol kis adatmennyiséget kell átvinni)? Mi szól a TCP mellett? Indokolja a válaszait!

   Megbízhatóság:

   - **TCP megbízható** kapcsolatot biztosít. Ez azt jelenti, hogy az adatokat az eredeti **sorrendben és hibamentesen** küldi át a forrás a címzetthez. Ha valamilyen hiba vagy adatvesztés következik be, a TCP újraküldi az elveszett vagy sérült adatokat. Ez különösen fontos olyan alkalmazásoknál, mint az HTTP, ahol a teljes, hibamentes adatátvitel elengedhetetlen. **Hibakezelés**
   - Az UDP **nem** garantálja a **megbízható adatátvitelt**. Nincs kézbesítési megerősítés vagy újraküldési mechanizmus, tehát az adatokat a címzettnél a sorrend vagy hibák nélkül átviheti, de ezzel **nő a lehetőség az adatvesztésre**.

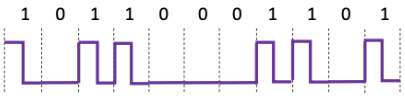
   Összességében az **UDP gyorsabb** és kevesebb hálózati overheadet jelent, de megbízhatóságban és adatintegritásban gyengébb, míg a **TCP megbízható kapcsolatot** biztosít, de **több hálózati overheaddel** jár.

2. Adatátvitel esetén milyen problémát okozhat a küldő és a fogadó óráinak elcsúszása? Megoldja-ezt a problémát az RZ kódolás(elején: 1-es bit magas jelszint/ 0-s bit alacsony jelszint, közepén: 1-es bit esetén váltás, végén: semmi)? Milyen kódolást alkalmaznak a 10 Mbps és 100 Mbps Ethernet szabványok? Milyen hátrányai vannak ennek a megoldásnak? Indokolja válaszát!

   Ha elcsúsznak az órák, akkor elmosódnak a bithatárok, nehéz eldönteni, hogy például egy magas jelszint az az előző, vagy az aktuális bit része lesz, hiba csúszhat a kommunikációba, illetve az apró késleltetés különbségek összeadódnak, egy idő után **összejön egy teljes ciklusnyi idő, ekkor egy bit kimarad**, vagy kétszer vesszük figyelembe azt.

   RZ nem oldja meg, mivel, ha a küldő oldalon egy hosszú, azonos jelekből álló sorozatot küld, akkor ezzel a vevő deszinkronizációhoz vezethet.

   RZ (1-es bit magas jelszint/ 0-s bit alacsony jelszint, 1-es bit esetén váltás, semmi)

   10 Mbps Ethernet **Manchaster** kódot használ, minden adatbitet két ciklussal fed le, a jelszint átmenettei határozzák meg a bitet => a sávszélesség a valódi bitráta felé. (Manchester- semmi, 1-es bit magasról alacsonyra/ 0-s alacsonyról magasra, semmi). Megoldás az órák elcsúszásának problémájára, viszont negatívum, hogy az átvitel felét használja ki (két óraidő ciklus per bit = 50%-ot romlik a hatékonyság)

   Manchester (semmi, 1-es bit magasról alacsonyra/ 0-s alacsonyról magasra, semmi)

   100 Mbps **4 / 5 bit NRZI** kódolást használ, a lényege, hogy 4 bitenként 1 bitet szúr be úgy, hogy limitálja a bitsorozat elején és végén a 0-k számát. Biztosan lesz jelszint változás, amihez igazíthatjuk az órákat, és csak 20%-ot romlik a hatékonyság.

3. A különböző autonóm rendszereken belül inkompatibilis routing protokollok (távolság vektor vs kapcs.állapot alapú) is használhatók. Ennek ellenére a globális forgalomirányítás működik. Hogyan lehetséges ez?

Távolság vektor: minden router-nek egy táblázatot kell karbantartania, amelyben minden célhoz szerepel a legrövidebb ismert távolság és annak a vonalnak az azonosítója. A táblázatot a **szomszédoktól származó információk** alapján frissítik. **Elosztott Bellman-Ford**(**jó hír gyorsan, rossz lassan terjed**) forgalomirányítási algoritmusként is nevezik. **Kapcsolatállapot alapú forgalomirányítás**: (**Dijkstra** algoritmus) nagy hálózatoknál számítás költséges és memória igényes. Minden router rálátást (link-state) gyűjt a hálózatról, és ezt az információt a teljes topológiáról osztja meg más routerekkel.

A globális forgalomirányítás működése annak köszönhető, hogy az autonóm rendszerek között közös protokollokat használnak a forgalomirányítás átjárásához. A Border Gateway Protocol (BGP) a legelterjedtebb forgalomirányítási protokoll, amelyet az autonóm rendszerek között használnak. A BGP segítségével az autonóm rendszerek kommunikálnak egymással, és megosztják az útvonalinformációkat, hogy megtudják, hogyan jutnak el az adatok egy adott hálózatból egy másikba.

4. Hasonlítsa össze a távolságvektor alapú (distance vector) és a kapcsolatállapot alapú (link-state) routing protokollokat? Mik a fő különbségek? Eldönthető-e, hogy melyik a jobb? Miért? Indokolja avválaszát!

Távolság vektor: minden routernek egy **táblázatot kell karbantartania**, amelyben minden célhoz szerepel a legrövidebb ismert távolság, és annaka vonalnak az azonosítója, amelyikhez el lehet jutni. A **táblázatot a szomszédoktól származó információk alapján frissítik**. Elosztott **Bellman-Ford** forgalomirányítási algoritmusként is nevezik (**jó hír gyorsan, rossz lassan terjed**). **ARPANET** eredeti forgalomirányító algoritmusa volt.
• 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.

Kapcsolatállapot alapú forgalomirányítás: A kapcsolatállapot alapú protokollok **minden routernek egy teljes képet adnak a hálózat topológiájáról**. A routerek rendszeresen küldenek ki üzeneteket, amelyek tartalmazzák a saját állapotukat, a csatlakoztatott linkek költségét és a szomszédos routerek azonosítóját. A routerek ezeket az üzeneteket továbbítják a többi routernek, így minden router ugyanazt a kapcsolatállapot adatbázist építi fel. A kapcsolatállapotot úgy is elképzelhetjük, **mint egy térképet**, ami megmutatja, hogy a hálózat milyen csomópontokból és linkekből áll, és mennyibe kerül áthaladni rajtuk.
(**Dijkstra** algoritmus) **nagy hálózatoknál számítás költséges és memória igényes**. Minden router rálátást (link-state) gyűjt a hálózatról, és ezt az információt a teljes topológiáról osztja meg más routerekkel.
AZ ALAPÖTLET ÖT LÉPÉSBŐL TEVŐDIK ÖSSZE
1. Szomszédok felkutatása, és hálózati címeik meghatározása.
2. Megmérni a késleltetést vagy költséget minden szo mszédhoz.
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.
• A kapcsolatállapot tartalma: a feladó azonosítója**, egy sorszám**, egy korérték és a szomszédok listája.

Összehasonlítás:
1. Méret és skálázhatóság:
    a. **Kapcsolatállapot**: **Általában hatékonyabb nagyobb hálózatokban** is, mivel a konvergencia gyorsabb és a skálázhatóság jobb.

5. Átviteli forgalom:

a. Távolságvektor: **Periodikus frissítések** miatt az átviteli forgalom magasabb lehet.
b. Kapcsolatállapot: Csak akkor cserélnek információkat, **ha változás történik**, ami kevesebb átviteli forgalmat eredményez.

6. Robusztusság és hibakezelés:
   a. Kapcsolatállapot: Általában jobb robusztusságot és **hibakezelést** biztosít.

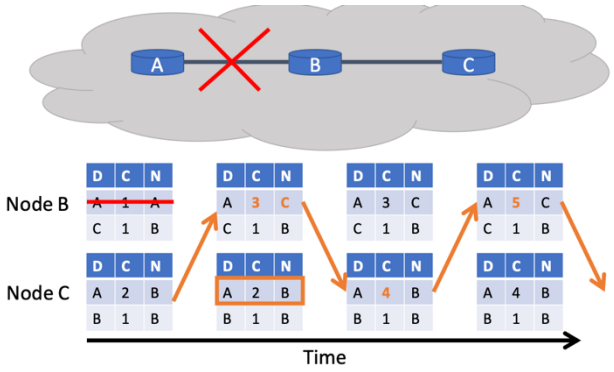
A választás a jobb protokolltól függ a konkrét alkalmazási követelményektől és a hálózati környezettől.
Kis méretű, egyszerű hálózatokban -> távolságvektor protokoll
Nagyobb, összetettebb hálózatokban -> kapcsolatállapot protokoll

A távolságvektor alapú protokollok egyszerűbbek és kevesebb memóriát és sávszélességet igényelnek, de lassabban konvergálnak és nem tudnak kezelni a negatív költségű hurkokat.

A kapcsolatállapot alapú protokollok gyorsabban konvergálnak és jobban alkalmazkodnak a hálózat változásaihoz, de több memóriát és sávszélességet igényelnek, és nehezebbek megvalósítani.
A protokoll választása függ a hálózat méretétől, forgalmától, dinamikusságától és biztonsági követelményeitől.

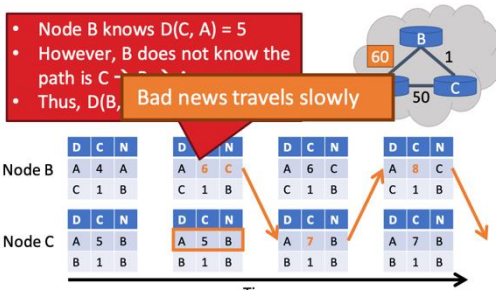
5. Mi az a count-to-infinity (végtelenig számlálás) probléma? Mi okozza? Mi történik a hálózat működése szempontjából, ha nem kezeljük? Hogyan oldjuk meg? BGP-nél látott routing protokoll esetén miért nincs jelen?

Bellman-Ford(jó hír gyorsan, rossz lassan terjed) távolságvektor alapú forgalomirányítási algoritmusnál fordul elő a rossz hír terjedésénél a count-to-infinity probléma. Rossz hír lehet egy csomópont megszűnése vagy csomópont költsége megnő. Oka, hogy router-ek topológiaváltozás esetén nem egyidőben változtatják táblázataikat, hanem a kommunikációval eltöltött idő miatt késleltetve. A hatása az lesz, hogy a végtelenségig emelkedik a költség. A számolás ideje alatt azonban hurok keletkezik, a csomagokat A és B egymásnak küldözgeti, amíg azok élettartama le nem jár. Ez hatalmas torlódást okoz és számos csomag célba sem ér.



A rossz hír terjedésénél végtelen ciklusok keletkezhetnek, ezért a megadott konstansig(=infinity) nő a költség, amikor leáll.

Megoldás a „split horizon with poison reversed”, azaz a router azt az információt, amit X szomszédjától hallott, nem küldi vissza X-nek, vagyis egy negatív információt küld vissza(végtelen költséget).



A BGP (Border Gateway Protocol) egy olyan routing protokoll, amely nem használ távolságvektort, hanem útvonalvektort. Ez azt jelenti, hogy a routerek **nem csak a költséget, hanem az útvonal teljes útvonalát is hirdetik a többi routernek**. Ez lehetővé teszi, hogy a routerek felismerjék és **elkerüljék a routing loopokat**, mert ha egy router látja a saját azonosítóját egy útvonalban, akkor tudja, hogy az egy loop.

BGP esetén nem csak azt hírdetik, hogy kik érhetőek el, hanem a teljes útvonalat, ami egy cél felé használnak. Ekkor a routerek képesek felépíteni a teljes gráfot, amin rögtön látszik, hogy merre van hurok, illetve, hogy kiesett egy eszköz.

6. A statikus csatorna hozzáférésnél látott módszerek közül, mi a módszerek hátránya? Miért nem megfelelőek ezek a módszerek számítógépes hálózatok esetén? Fennállnak-e ezek a hátrányok a CDMA módszernél? Miért? Indokolja válaszát!

Mindegyik módszer felosztja a csatornát n résztvevőre, ez által **nem garantált a folyamatos küldés**, így sűrűn előfordulhat, hogy csak részlegesen van kihasználva a csatorna, ami a hatákonyságot rontja. Pl egy folyamatosan küld, a többi meg néha néha küld vlmit, így **kihasználatlan** lesz rendszer.

A statikus csatorna hozzáférés olyan módszer, amely előre meghatározza, hogy mely felhasználók használhatják a közös csatornát, és mennyi időre. Ilyen módszerek például a **frekvenciaosztás** (FDM), az **időosztás** (TDM-Time Division Multiplexing) vagy a **polarizációosztás** (PDM), térbeli multiplexálás (SDM), **hullámhossz** multiplexálás (WDM)

A statikus csatorna hozzáférésnek több **hátránya** is van, különösen **löketes forgalom** esetén, amikor a felhasználók nem küldenek folyamatosan adatokat, hanem időnként nagyobb adagokat. Ilyenkor a statikus **csatorna hozzáférés pazarló, mert nem használja ki a csatorna teljes kapacitását, és nem alkalmazkodik a forgalom változásaihoz.**

A számítógépes hálózatok általában löketes forgalmat bonyolítanak, ezért a statikus csatorna hozzáférés nem megfelelő választás. A számítógépes hálózatoknak dinamikus csatorna hozzáférésre van szükségük, amely lehetővé teszi, hogy a felhasználók akkor használják a csatornát, amikor szükségük van rá, és a csatorna kihasználtsága magas legyen.

A CDMA (Code Division Multiple Access) egy **dinamikus** (ortogonális) csatorna hozzáférési módszer, amely több felhasználónak teszi lehetővé, hogy egyszerre használják a csatornát, anélkül, hogy zavarják egymást. A CDMA módszer előnyei a következők:

- **Nagyobb kapacitás**, mert a felhasználók nem osztoznak a csatorna sávszélességén, hanem a teljes sávszélességet használják
- **Jobb minőség**, mert a CDMA kódolási technikája csökkenti a zajt és az interferenciát
- **Rugalmas erőforrás-elosztás**, mert a CDMA nem igényel időzítést vagy koordinációt a felhasználók között
- **Alacsonyabb teljesítmény-igény**, mert a CDMA alacsonyabb jel-erősséggel is működik

7. Hasonlítsa össze a iteratív és rekurzív DNS lekérdezéseket! Mi az előnye és a hátránya ezeknek? Egy új domain név bejegyzése után (új entry): miért fordulhat elő az, hogy egyik helyről elérhető a szerver (sikerül a névfeloldás), míg másik helyről nem, és sokszor 72 órát kell várni arra, hogy mindenhonnan feloldható legyen az újonnan bejegyzett név?

A **rekurzív** névfeldolgozás akkor következik be, amikor egy DNS szever nem ismeri a kersett domaint, de **tudja kihez kell érte fordulni**. Ekkor ezt meg is teszi, és ha megkapja az infót, akkor ő fog válaszolni az eredeti kérdezőnek. Ennek hátránya, hogy **nem szolgálható ki azonnal a kérés, meg kell várni a választ, addig is tárolni kell, hogy kinek kell válaszolnia majd => erőforrás igényesebb.**

Iteratív esetben a DNS szerver megmondja, hogy nem ismeri a domaint, de tudja **kit lehetne megkérdezni**. Ezt továbbítja a kérdezőnek. Innentől a kérdező feladata megszólítani a másik/többi DNS szervert is az infókért. **Ez leveszi a terhet a DNS szerverekről, de plusz terhelés a felhasználó eszközének.**

Azért lehetséges, hogy 72 órát kell várni, mert az összes lokális névszerveren le kell, hogy frissüljön a bejegyzett domain.

Az új rekord bevezetésekor rögzítik a bejegyzést egy szerveren, de egy másik DNS szerver elképzelhető, hogy **cachelte ennek a szervernek a bejegyzéseit**. Ennek a cachenek a **lejárati ideje szokott lenni a 72 óra**. Mikor a másik szerverre érkezik be a kérés, látja, hogy tud mindent az eredeti DNS szervertől, neki nincs ilyen bejegyzése (a cache alapján), így ezzel a válasszal tér vissza, pedig idő közben ez a bejegyzés bekerült.

8. Hasonlítsa össze az előadáson látott 3 CSMA variánst. Melyik a jobb? Milyen feltétel mellett? Mit mondhatunk a késleétetésről? Indokolja a válaszait!

CSMA- Carrier Sense Multiple Access

1-perzisztens

Időmodell folytonos. Belehallgat a csatornába, ha foglalt akkor vár amíg szabad lesz.
Jobb teljesítmény mint az ALOHA protokollok, viszont generálja az ütközést.

Nem-perzisztens

Időmodell folytonos. Mohóság elkerülése. Belehallgat a csatornába, ha foglalt akkor véletlen ideig várakozunk, majg kedi előről a köldési algoritmus.
Jobb teljesítmény mint az 1-perzisztens CSMA protokoll (sok mindentől függ).

P-perzisztens

2 közötti, idő slotok vannak.
Belehallgat a csatornába, ha foglalt akkor a kövi idő csatornában újra belehallgat. Ha szabad,
Akkor p valószínűséggel küld.
Ha nagyon nagy a terhelés, akkor ezt el tudja látni.

Ha kicsi a terhelés, akkor 1p protokoll jobb, azonnal küldi amint szabad, nem lesz várakozás.
Nagy terhelésnél, akkor a nem-p jobb, mert az 1p mindig belehallgat, de foglalt. Várhatóan van még x állomás, aki vár, ha felszabadul a csatorna akkor mindenki egyszerre küld, garantált ütközés.

9. Hol kerül először feldolgozásra a forrás egy adott rétege által a csomagra rakott információ (fejléc)? Mi a helyzet TCP és IPv4 protokollok esetén? Mely az előadáson látott hálózati funkció(k) törik meg az Internet ezen tiszta modelljét?

Megoldás:

10. Mi a lényege az ISO/OSI rétegmodellnek? Milyen problémát old meg? Milyen rétegeket definiál a modell és ezek közül melyekkel írható le az Internet működése?

OSI- Open System Interconnect Model.

- Az ISO/OSI rétegmodell egy referencia modell a számítógépes hálózati protokollok tervezéséhez és értelmezéséhez, 7 absztrakciós rétegben. A modell célja, hogy szabványosított módon definiálja a különböző hálózati funkciókat és

rétegeket, amelyek összehangoltan működnek, hogy lehetővé tegyék a hálózatok hatékony működését és a különböző rendszerek közötti kommunikációt. Az ISO/OSI rétegmodell az International Organization for Standardization (ISO) által kifejlesztett és elfogadott.

- A rétegmodell segíti a fejlesztőket és tervezőket a hálózati protokollok tervezésében és megvalósításában, mivel a különböző rétegek elválasztottak, így könnyebb azokat külön fejleszteni és karbantartani.

7.Alkalmazási-üzenetek, BitTorrent, HTTP, BitCoin

6.Megjelenítési- adatkonverzió

5.Ülés- szinkronizációs pont

4.Szállítói-szegmens, üzenetek adott állomáson belüli forgalom multiplexálása, TCP és UDP

3.Hálózati-csomagok, csomagtovábbítás, útvonal választása

2.Adatkapcsolati-framek, átvitelre szánt adatok keretekre tördelése, közeghozzáférés vezérlése (MAC)

1.Fizikai-bitek, optikai kábel, bitek, Wifi jel, CAT6UPT

Az Internet működését az OSI modell négy rétegével írhatjuk le, amelyek a következők:

1. Fizikai réteg
2. Adatkapcsolati réteg
3. Hálozati réteg
4. Szállítási réteg
5. Megjelenítési

11. Miért nem alkalmas a switcheknél látott csomagtovábbítás ás MAC cím tanulás globális forgalomirányításra? Miért van szükség az IP-re? Indokolja a válaszát!

A switcheknél látott csomagtovábbítás és MAC (Media Access Control) cím tanulás **nem alkalmas globális forgalomirányításra**, mert ezek a módszerek csak a helyi hálózaton belül működnek. A **switchek** nem tudják meghatározni, hogy egy adott célállomás melyik hálózaton található, és hogyan lehet elérni azt. A switchek csak a MAC címek alapján továbbítják a kereteket, amelyek nem egyértelműen azonosítják a hálózati eszközöket.

Az IP-re azért van szükség, mert ez a protokoll lehetővé teszi, hogy a **különböző hálózatok között kommunikáljanak** az eszközök. Az IP egy **logikai címzési** rendszert használ, amely megkülönbözteti a hálózatokat és az azokhoz tartozó állomásokat. Az IP segítségével a routerek meghatározhatják, hogy melyik **útvonalon** továbbítsák a csomagokat a forrás és a cél között. Az IP a hálózati réteg protokollja, amely független a fizikai és az adatkapcsolati réteg protokolljaitól.

12. Milyen előnye van a szélessávú átvitelnek az alapsávú átvitelhez képest? Milyen modulációs techikákat ismer? Tegyük fel, hogy egy cos(t) függvény a vivőhullám és a s(t) függvény az elküldő jel. Ezek közül melyek azok, amelyek adatátvitelre használhatók? Mi a különbség az analóg és digitális jelek modulációja között?

A szélessávú átvitelnek több előnye is van az alapsávú átvitelhez képest, például:

- **Frekvenciatartomány kihasználása, nagyobb sávszélesség**, mert több jelet küldhetünk egyszerre különböző frekvenciákon egyetlen csatornán keresztül

- **Jobb zajvédelem**, a szélessávú jelek kevésbé érzékenyek az interferenciára és zajokra, mivel a több frekvencia átlagolja ezek hatását
- **Több felhasználó támogatása**, mert a frekvenciaosztásos multiplexelés lehetővé teszi, hogy különböző jelek megosszák ugyanazt a csatornát anélkül, hogy zavarnák egymást

Alapsáv-baseband Szélűessáv-broadband

A modulációs technikák olyan eljárások, amelyek segítségével egy vivőjel képes információt hordozni. A modulációs technikák két nagy csoportra oszthatók: analóg és digitális.

Amplitúdó moduláció, Frekvencia moduláció, Fázis moduláció

Digitális átvitel – Diszkrét szignálok véges halmazát használja (például feszültség vagy áramerősség értékek).

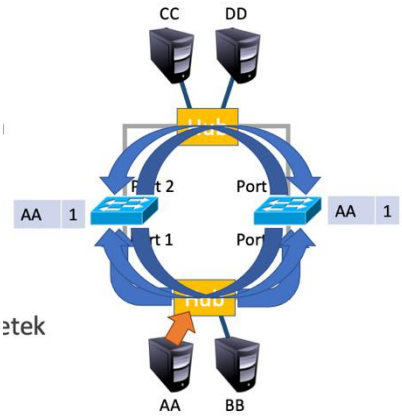
Analóg átvitel– Szignálok folytonos halmazát használja (például feszültség vagy áramerősség a vezetékben)

• Digitális előnyei- Lehetőség van a vételpontosság helyreállítására, illetve az eredeti jel helyreállítására

• Analóg hátránya- A fellépő hibák önmagukat erősíthetik

13. Milyen problémát old meg az STP (Spanning Tree Protocol) - Feszítőfa Protokoll? Ha egy hálózatban eltérő kapacitású linkek is vannak, akkor milyen problémát okozhat az STP? Gondoljon a hatékonyságra!

Az STP Feszítőfa Protokoll az **Ethernet** hálózatokban alkalmazott protokoll, és célja a **hurokmentes topológiát** biztosítani a hálózati útvonalakon keresztül. Gyenge a teljesítménye, mert a feszítőfa nem foglalkozik a terhelés elosztással. Nem hatékony és gyenge skálázhatóság.
Kezdetben minden elem a gyökérelem.



Ha egy hálózatban eltérő kapacitású linkek is vannak, akkor az STP problémát okozhat a hatékonyság szempontjából. Az STP ugyanis nem veszi figyelembe a linkek költségét vagy sávszélességét, amikor a feszítőfát kialakítja, hanem csak a kapcsolók prioritását és azonosítóját. Ez azt jelenti, hogy **az STP nem feltétlenül választja a legjobb útvonalat a hálózatban**, és nem használja ki a rendelkezésre álló erőforrásokat.

- Lefed minden csomópontot
- Nem tartalmaz köröket
- Nem választja a legjobb utakat -> nem hatékony
- Nem foglalkozik a terheléses elosztással
- Hot spots
- Gyenge skálázhatóság

14. Mire szolgálnak a HTTP cookie-k (sütik)? Tegyük fel, hogy egy népszerű közösségi hálózat felhasználója (nevezzük FB-nek) vagyunk. Hogyan tudja elérni az FB HTTP Cookie-k (sütik) segítségével, hogy információt kapjon az általunk olvasott tartalmakról (pusztán egy oldal meglátogatásával, külön felhasználói interakció nélkül)?

Kosarak, személyreszabás, bejelentkezés, célzott hírdetések.

A HTTP cookie-k (sütik) olyan kisméretű információcsomagok, amelyeket a weboldalak küldenek a böngészőnek, és amelyeket a böngésző visszaküld a weboldalaknak minden kérés alkalmával. A cookie-k

segítségével a weboldalak megjegyezhetik a felhasználók beállításait, bejelentkezési adatait, böngészési előzményeit és egyéb információkat, amelyek javítják az online élményt.
Felhasználás:

- Állapotkezelés- webhelyek tudják, mikor egy felhasználó be van jelentkezve vagy mik a preferenciái
- Nyomonkövetés- Felhasználhatók a felhasználói tevékenység nyomkövetésére, statisztikák gyűjtésére, és célzott hirdetések szolgáltatására.
- Személyre szabott tartalom

Ha egy népszerű közösségi hálózat (FB) felhasználója vagyunk, akkor az FB cookie-kat használhat arra, hogy információt kapjon az általunk olvasott tartalmakról. A böngésző azonosítja a felhasználót és a weboldalt. Ezután a böngésző visszaküldheti ezt a cookie-t az FB-nak, amikor a weboldalról kérést küld. Így az FB nyomon követheti, hogy milyen weboldalakat látogatunk meg, és milyen tartalmakat olvasunk.

15. Mire szolgál a TCP slow start mechanizmusa? Mennyire lassú? Milyen adatfolyamoknál okoz ez gondot a teljesítményben (főleg nagy sebességű hálózatoknál)? Milyen megoldási javaslatok születtek erre?

Cwnd- congestion window (torlódási ablak, hálózat képessége)

Ssthresh= Adv_wnd- advertise window (fogadó képessége)

Ssthresh- küszöbérték

A TCP slow start mechanizmus **a legszűkebb sávszélesség** meghatározására szolgál alacsonyabb sebességű hálozatokon. Cél, **hogy gyorsan elérjük a könyök pontot**. A slow start valójában **nem lassú(exponenciális)**. Congestion window(cwnd) gyorsan nő, majd lelassul amikor csomagvesztés történik.

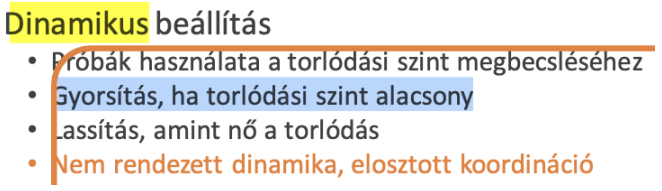
A TCP slow start mechanizmusa egy olyan algoritmus, amely a TCP csomagok küldésének sebességét a hálózat forgalmának függvényében állítja be. A célja, hogy **elkerülje a hálózat túlterhelését**, és **optimalizálja a sávszélesség kihasználását.**

A TCP slow start mechanizmusa úgy működik, hogy a TCP kapcsolat kezdetén a küldő fél egy kis mennyiségű adatot küld, amelyet a fogadó fél visszaigazol. Minden visszaigazolás után a küldő fél exponenciálisan növeli a küldhető adatok mennyiségét, amíg el nem éri a hálózat kapacitását, vagy amíg nem tapasztal csomagvesztést. Ha csomagvesztés történik, akkor a küldő fél csökkenti a küldési sebességet, és újra lassan növeli azt.

Gond azok az adatfolyamok, amelyek **rövid ideig tartanak** vagy amelyek **nagy mennyiségű adatot küldenek**. Ilyen adatfolyamok lehetnek például a weboldalak letöltése, a videók streamelése, vagy a nagy fájlok átvitele.

Megoldások:

- A küldési sebesség növelésének gyorsítása a slow start fázisban, például a kezdeti ablakméret növelésével, vagy a visszaigazolások számának növelésével
- A küldési sebesség csökkentésének mérséklése a csomagvesztés esetén, például a csomagvesztés okának megkülönböztetésével, vagy a küszöbérték dinamikus beállításával



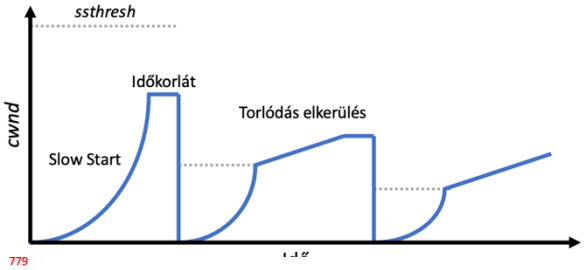
- A küldési sebesség adaptív beállítása a hálózat állapotának függvényében, például a késleltetés, a sávszélesség, vagy a torlódás mérésével, vagy a visszaigazolásokban lévő visszacsatolási információk használatával.
- Sütiben tárolni az előző sávszélesség méretét

16. Mutassa be a fő különbséget a hagyományos TCP torlódásvezérlés (pl. TCP Tahoe vagy Reno vagy CUBIC, stb.) és a DCTCP között? Elérhető-e fair erőforrás megosztás egy hagyományos és egy DCTCP folyam között? Miért igen vagy miért nem? Gondoljon a torlódásra adott válasz reakcióra! Indokolja válaszát!

TCP: Két résztvevő interneten, ahol egy résztvevőt egy IP-cím és egy port azonosít, AIMD (Additive increase, multiplicative decrease), globális használat. Torlódás-hálózat nagyobb terhelése mint a kapacitása

DCTCP: adatcenter, ECN használata, agresszív torlódásvezérlés

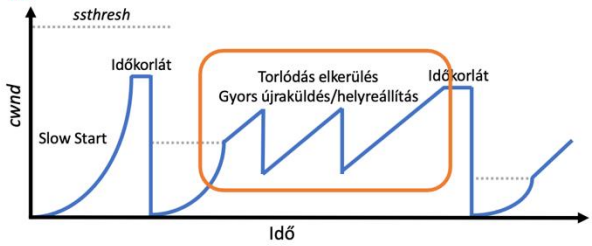
A hagyományos TCP torlódásvezérlés úgy mükődik, hogy ha csomag vesztést tapasztal vissza vesszi az avlakméretét, ezáltal csökken a sebesség. Az említett hagyományos protokollok abban különböznek, hogy miként vesznek vissza a sebességből, majd ezután hogyan gyorsítanak fel újra. Költséges a csomagog újraküldése, de a másik probléma, hogy ingadozik a sávszélesség, ami miatt kihasználatlan az erőforrás.



Cwnd- congestion window (torlódási ablak, hálózat képessége)

Ssthresh= Adv_wnd- advertise window (fogadó képessége)

Ssthresh- küszöbérték



• Stabil állapotban, a cwnd az optimális ablakméret körül oszcillál

TAHOE- 1974, torlódásvezérlés

TCP Reno- TAHOE+:

1. nem jön nyugta, idő tullépés-> vissza slow startba (mint a TAHOE)

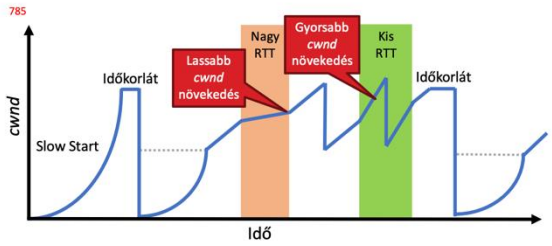
2. gyors újraküldés(3 duplikált ACK-> újraküldés) -> gyors helyreállítás(csomagvesztés esetén: set cwnd= cwnd/2 **(nem slow starba kell visszamenni mint a TAHOE-nel)** ->optimális ablakméret körül oszcillál. ------>

New Reno- Reno + javít a gyors újraküldésen (nem kell 3 duplikált nyugta, viszont ha a sorrend rossz akkor újabb probléma...)

TCP-nél mindig lesznek csomagvesztések...

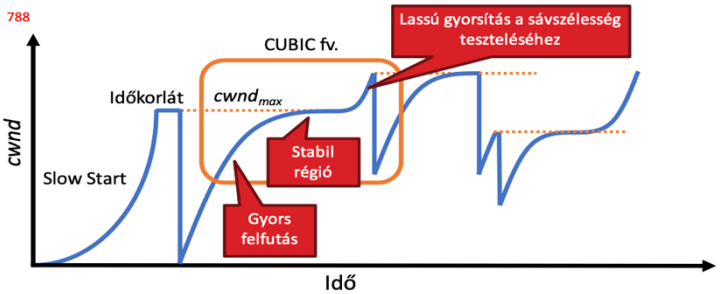
Compound TCP (CTCP) Windows
Reno alapú
2 torlódási ablak(késleltés/ vesztés alapú)
Küldési ablak-min(cwnd + dwnd, adv_wnd)

Compound TCP példa



• Agresszívan reagál az RTT változására
• Előnyök: Gyors felfutás, sokkal fairebb viselkedés más folyamokkal szemben eltérő RTT esetén
• Hátrányok: folyamatos RTT becslés

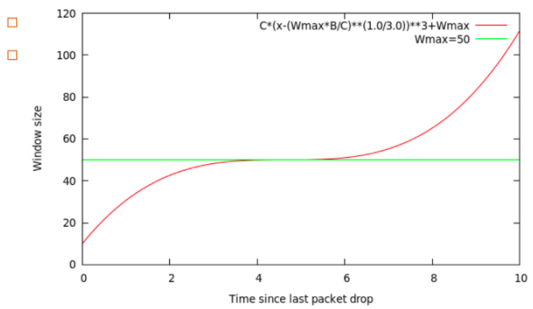
TCP CUBIC példa



• Kevésbé pazarolja a sávszélességet a gyors felfutások miatt
• A stabil régió és a lassú gyorsítás segít a fairség biztosításában
  • A gyors felfutás sokkal agresszívabb, mint az additive increase
  • A Tahoe/Reno variánsokkal szembeni fairséghez a CUBIC-nak nem szabad ennyire agresszívnak lennie

TCP Cubic (Linux)
AIMD helyettesítése köbös függvénnyel

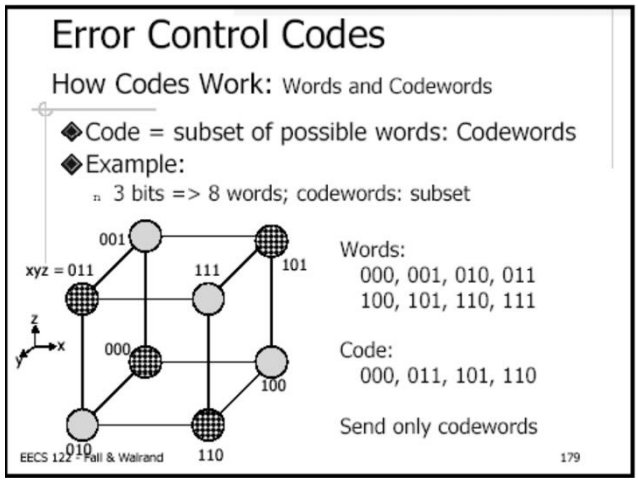
TCP CUBIC



DCTCP-Data Center TCP
Adat központokhoz találtak ki, ECN használata- arányosan változtat/ vesz vissza a torlódási ablakon (jobb hatékonyság!!)
Adatközpontokban nagyon jól működik, sokkal jobban ki lehet használni a sávszélességet

---

17. Tegyük fel, hogy 10 bit hosszú kereteket küldünk és olyan kódolást használunk, ami 3 bithiba javítására képes. Adjon felső korlátot a legális kódszavak (valid keretek) halmazának méretére? A 10 bites keretből legfeljebb hány bit lehet a hasznos adatbitek száma és mekkora a redundáns rész? Indokolja válaszá.!



N= m + r
N – bit hossza
R – redundáns bitek (ellenőrző bitek)
M – üzenet bitek

7 bit hasznos, 3 redundáns

18. Tegyük fel, hogy minden keretbe 1000 bájt hasznos adat fér (Megj.: a keretezés után a keret mérete lehet más, de a legrosszabb esetben is befér 1000 adatbájt a keretbe.). Az előadáson látott karakterszámlálás, bájt beszúrás és bit beszúrás módszerek esetén mekkora teljesítmény csökkenést kapunk a legjobb és a legrosszabb esetben? Indokolja a választát példák segítségével!

Keretekre tördelésre azért van szükség, mert a fizikai réteg nem garantálja a hibamentességet. Az adatkapcsolati réteg feladata a hibajelzés illetve a szükséges szerinti javítás. A keretezés nem egyszerű, mivel megbízható időzítésre nincs lehetőség.

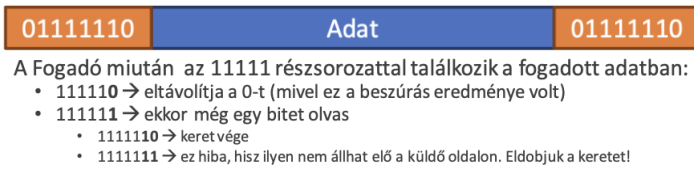
Karakterszámlálás
-nagyon érzékeny a hibára – nem jó

Bájt alapú beszúrás (gyakorlatban: DSL, cellular)
-jó megoldás
-speciális FLAG bájt jelzi az adat két végét (szükséges az bájt is ha a flag az adatban lépne fel)
-50% a teljesítmény csökkenés, legjobb 0%

- Egy speciális **FLAG** bájt (jelölő bájt) jelzi az adat keret elejét és végét

ESC

Bit beszúrás (gyakorlatban: HDLC)
-jó megoldás
-flag helyet x db bit jelzi a két végét
-az adatban minden x-1 hosszú sorozat után beszúrunk “0” bitet



egy

-legrosszabb esetben 20% teljesítmény csökkenés, legjobb esetben 0%

19. Több rétegben is találkozhatunk a csúszóablak protokollal. Sok esetben kumulatív nyugtát alkalmazunk, azaz a nyugta-sorszám a következő várt adategységet azonosítja és egyben az összes megelőzőt nyugtázza. Miért jobb ez, mint minden sorszámra egyedi nyugta küldése (azaz

egyedi nyugtázás)? Milyen esetben előnyösebb a kumultatív megoldás?

kumulatív nyugta– Olyan nyugta, amely **több keretet nyugtáz egyszerre**. Például, ha a 2,3 és 4 kereteket is fogadnánk, akkor a nyugtát 5 sorszám tartalommal küldenénk, amely nyugtázza mind a három keretet. Ha viszont a 3-as keret nem jött meg, akkor ezt nyugtázza, majd miután a küldő vissza küldte akkor a már 4-es meg van, szóval az 5-öt nyugtázza.

Hátránya akkor van, mikor 1 csomag veszett el, és az utána lévők célba értek. Mivel a feladó nem tudja, hogy a későbbi csomagok rendben voltak, így mikor észre veszi, hogy az elveszett csomagot újra kell küldenie, valószínűleg küldeni fogja az utána jövőket is, amit nem lenne szükséges, mert azok sikeresen célba értek.
→**Generálhat felesleges újraküldéseket timeout miatt**. **Nem ideális olyankor, ha gyakori a csomagvesztés.**

Kumulatív nyugta előnyei:
-kisebb nyugtaszám -> sávszélesség és feldolgozási idő javul
-egyszerűbb megvalósítás
-kevesebb memória

Egyedi nyugta előnyei:
-pontosabb hibakezelés

20. Adott egy fizikai berendezés, ami rögzített sebességgel képes szimbólumokat a hálózatra írni. Feltéve, hogy más korlátozás nincs, milyen technikával tudjuk megduplázni az adatrátáját (bps) ennek a rendszernek? Gyakorlatban ez milyen problémát jelent? Indokolja a válaszát!

Szimbólumok bevezetése:
-A(0,0) B(1,0) C(0,1) D(1,1). Egy szimbólummal 2 bitet vittünk át.
Elején elküldeni a szimbólumokat? Ha kevés adat van nem éri meg

21. Mi az oka, hogy a réselt ALOHA jobb teljesítményre képes mint a sima ALOHA? Indokolja a válaszát!

Megoldás:
Azért mert a réselt Aloha az időrések határán hajtja végre az átvitelt, amíg a sima Aloha véletlen ideig vár az átvitellel.
Ezért hosszú távon biztosabb a réselt Aloha alkalmazása, mert kiszámíthatóbb a
Működés
6.ea eleje

Sima ALOHA-1970, havan elküldendő adat akkor elküldi. Alacsony költségű, de egyszerű megoldás. Állomás azonnal küld, fogadó mindent nyugtáz. Ha nincs nyugta, ütközés, vár majd újra küld. Minden keret azonos méretű, csatorna zajmente. Poisson folyam

22. Hogyan változik a sebezhetőségi idő az eredeti réselt ALOHA módszerhez képest, ha egy keret átvitele N résnyi időbe telik egy helyett. Átvitel csak rés határán történhet. A rések számában adja meg a sebezhetőségi időt! Indokolja a válaszát! Hogyan befolyásolja ez az átvitelt (S(G)) a terhelés függvényében (G)?