

Az IP jitter mentesítőt a mini PST-vel azonos vázba építettük, de igény esetén 19"-os rack kivitelben is szállítjuk majd



A tartalomból:

- Túltőkésítve
A cég átalakítása, megreformálása
- Switch és számítógép teszt
A teljesítőképesség és a működési mechanizmus vizsgálata
- Jitter mentesítés
Az IP Stream Tool bemutatása
- DVB T2-MI
Analizálás és a szállított tartalom kinyerése
- Új utasítás a PST szolgáltatásai között
A Real Time Analyzer csatornánkénti törlése

CableWorld

hírek

A CableWorld Kft. technikai magazinja
2020. június



Számunk fő témája:

Az IP Jitter megszüntetése

72.

Túltőkésítve !

Változások a CableWorld-nél

1993-ban a CableWorld működésének beindításánál alig tudtunk annyi pénzt összekaparni, hogy az első évet túléljük. Általában a kis cégek többsége kezdetben lényegesen kevesebb tőkével rendelkezik, mint amennyi a normális és biztonságos működéshez szükséges. Az ilyen cégekre akasztják az „alultőkésített” jelzőt, a CableWorld is ilyen volt.

A sikeres kezdőévek lehetővé tették, hogy a cég vagyona folyamatosan gyarapodjon. A rendszerváltás éveit követő ingatlan túlkínálatot kihasználva a cég 1996-ban már egy 2000 m²-es üzemépület megvásárlását is megengedhette magának. Igaz, hogy az épület üzemi területen, osztatlan telken számos hiányossággal rendelkezett, de az üzemeltetési költségek így is kisebbek voltak, mint a korábban bérelt területen.

A következő két évtized gondtalanul telt, a cég szépen fejlődött, a tulajdonosok elégedettek voltak az évente kifizetett osztalékkal, miközben a dolgozók is élvezték a kényelmes munkahelyekből származó előnyöket. A digitális technikára történő átállás piaca elegendően nagyra bizonyult, mivel a cég újszerű megoldásai iránt a világ összes országából érdeklődtek.

A nehézségek 2015-ben kezdtek jelentkezni, ugyanis a digitális átállás befejeződésével a korábbi piac töredékére zsugorodott. Számszerűsítve, az elérhető éves árbevétel közel tizedére csökkent, mivel az üzemeltetés már alig igényel új berendezéseket.

Időközben a tulajdonosi kör jelentősen megöregedett és nem tudta megérteni, hogy a korábban felhalmozott vagyon miért nem hoz annyi hasznot, mint korábban. A cég vezetése hiába adta ki a 2000 m²-es üzemépület felét más cégeknek, a bérleti díjból befolyt összeg meg sem közelítette a korábbi évek termeléséből adódó nyereséget. Mindezek mellett az általános rezsi költségek (az épület fenntartása, fűtés, villany, a cégvezetés költségei stb.) alig csökkentek, ami folyamatosan növelte a tulajdonosok elégedetlenségét.

Kívülről szemlélve elmondható, hogy a cég ekkorra már jelentős vagyonnal rendelkezett (üzemépület, autók, gépek, berendezések, készletek stb.), azonban az összezsugorodott piac nem teszi lehetővé, hogy a cég a tulajdonosok által elvárt nyereséget produkálja. Tanácsadó cégek ilyenkor javasolják, hogy profilt kell váltani, új piacokat kell keresni, de jelen esetben a bőven nyugdíjas korú tulajdonosoknak már nem érdeke a megújulás, nem tudják kivárni az évek múlva profitot hozó cégátalakítást.

A bevezetőben használt „alultőkésített” jelző mintájára, most a „túltőkésített” jelző használata illik a helyzet jellemzésére. Ilyenkor sokkal nagyobb a cég vagyona, mint ami a termeléshez, az elérhető éves árbevételhez szükséges. Könnyű mondani, hogy akkor csökkenteni kell a cég vagyonát, de hogyan?

Néhány évi stagnálást követően a hazai ingatlan piac jelentős mértékű élénkülése adott lehetőséget a vagyon csökkentésére. 2019 végére az épület iránt mutatkozó fokozott érdeklődés lehetővé tette a 2000 m²-es üzemépület eladását, ami a tulajdonosok megelégedését elnyerő árbevételt hozott. Az átalakítási folyamatban a használt, esetenként 15 ... 20 éves gépek és berendezések, valamint a készletek eladásából származó bevétel eltöri az ingatlan eladásából származó bevétel mellett.

Az ingatlan eladása és 2020. első felében történő kiürítése miatt a cégnek új telephely bérlése vált szükségessé. Mi sem jellemzőbb a méretek csökkentésére, minthogy a bérelt terület mindössze 2×60 m² nagyságú és így is megfelelő a tevékenység folytatásához. A cég telephelyének új címe:

1222 Budapest Nagytétényi út 100.

A költözés az új helyre viszonylag egyszerű, ugyanis alig néhány kilométerre van az eddigi telephelytől. A belakás, azaz a tevékenység indítása az új helyen terveink szerint 1 hónapot vesz igénybe. Ennél sokkal nagyobb gondot jelent az épület kiürítése, a sok-sok feleslegessé vált, de eddig ki nem dobott bútor, berendezés, anyag környezetkímélő elszállíttatása.

A feleslegessé vált mérőműszerek és berendezések értékesítésére februárban hirdetést adtunk fel. A börze keretében jelképes áron adtuk el az érdeklődőknek az oszcilloszkópokat, spektrum analízátorokat, szintmérőket, tápegységeket stb.. Érdemes megemlíteni, hogy miután az általános célú berendezések iránt mutatkozó érdeklődés elképzeléseinknek megfelelő volt, a digitális technika berendezései és egységei iránt szinte semmilyen érdeklődés nem mutatkozott. Ebből azt a következtetést vontuk le, hogy hazánkban jelenleg még igen szűk az a szakmai kör, aki bármit is tud csinálni (mérni, készüléket üzembe helyezni vagy programozni stb.) a digitális technika területén.

Cégünk az új helyen május végétől fogadja partnereit. Addig a korábbi telephelyen, vagy félúton a kettő között vagyunk megtalálhatók. Az informatika nyújtotta lehetőségeket kihasználva központi telefonszámunk változatlan.

A jövőt illetően még bennünk is számos kérdés vár megválaszolásra. A tőke lecsökkentése megtörtént, a tulajdonosok megkapták vagyonuk arányos részét, így a továbbiakban a cégnek nem kell foglalkozni a nagy vagyon után elvárt osztalék előteremtésében. Az átalakítás következtében a cég tőkésítettsége optimálisnak látszik. A kézzelfogható vagyon csökkent ugyan, de a tapasztalatok és a szakmai ismeretek halmaza nem csökkent, mivel az továbbra is mindössze 2-3 fő fejében van tárolva.

Switch és számítógép teszt

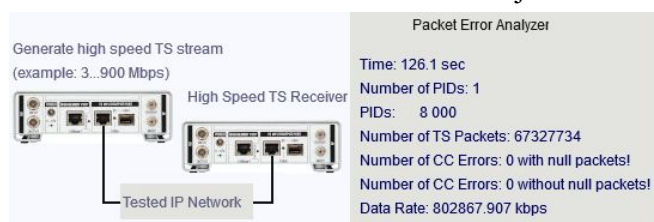
A teljesítőképesség és a működési mechanizmus vizsgálata

Előző számunkban egy célműszer fejlesztés keretében az IP jitter generálásával és mérésével foglalkoztunk. A cikk után többen is kérdéseket tettek fel a switch-ek és a PC-k teljesítő képességével kapcsolatban, ezért mielőtt elkezdenénk a jittermentesítéssel foglalkozni – részben saját magunk megnyugtatása érdekében – közzétesszük néhány mérésünk eredményét.

A különböző átviteli csatornák teljesítőképességének és hibamentességének vizsgálatához fejlesztőink a PST-be egy nagysebességű TS Generátort is beépítettek. A modullal 3 és 900 Mbps közötti tartományban végezhetünk méréseket. A készülék ténylegesen ilyen sebességű adatfolyamot állít elő, igaz, hogy ezt a modult használva a készülék többi funkciója kikapcsolásra kerül. A kimeneti adatfolyam CC számlálót is tartalmaz, így a packet vesztés könnyen kimutatható.

A v2.0 változattól a készülék egy speciális vevőt is tartalmaz, hogy a kiértékelés az átviteli csatorna végén ne jelentsen megoldhatatlan feladatot. A vevőegység megszakítás nélkül veszi az adatfolyamot, a szoftver 2 másodpercenként periodikusan kiolvassa és kijelzi az adatokat. A kijelzőn a vett packetek darabszáma, az eltelt idő és a CC hibák száma is látható.

Első feladatként a PST-vel állítsunk elő egy 800 Mbps körüli adatfolyamot, amelyet egy másik PST vesz és a kettő között csak egy CAT5-ös egyenes kábel van. A mérési összeállítást az 1. ábra mutatja.

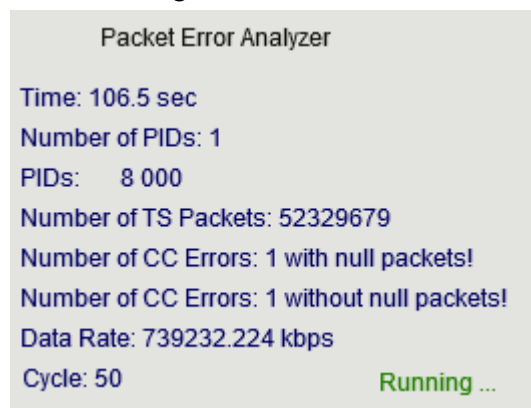


1. ábra

Mérési összeállítás nagy sebességű átviteli csatorna méréséhez

A kijelzőn látható, hogy a felvétel 2 perc elteltével készült, CC hiba nincs, de igazolni kellene, hogy a mérés valóban jelzi a hibákat. Az igazoláshoz húzzuk ki a kábelt majd dugjuk vissza. A rövid szünet miatt a kijelzett adatsebesség lecsökken, mert kevesebb TS packet érkezett és a visszadugásnál megjelenik az 1-es hibaszám. A tesztünk eredményét mutató mérőlapot a 2. ábra szemlélteti.

Elővettünk egy TP-Link 5 portos switch-et. Mivel ez nem multicastos switch, üzenetszórással küldte tovább a pecketeket, de hibát nem tapasztaltunk. Nehezítésként egy második, majd egy harmadik vevőt is csatlakoztattunk portjaihoz, de hibát továbbra sem tapasztaltunk.



2. ábra

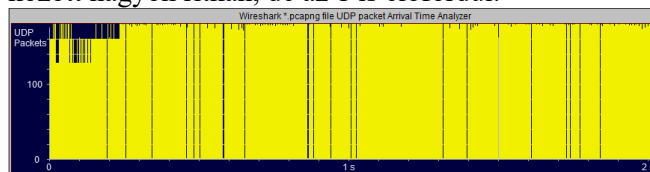
CC hiba előállítás a kábel kihúzásával, majd visszadugásával

A mérést multicastos Cisco gyártmányú switch-hel is megismételtük, de mint várható volt hibát nem találtunk az átvitelben.

Megjegyezzük, hogy a generátor csak közelítőleg tudja előállítani a kívánt sebességű adatfolyamot, mivel az időzítő nem tud a ns-ok tört részével dolgozni. A másik megjegyzendő, hogy a vevő más adatfolyamokat is fel tud dolgozni. Mivel a null packetekben nincs CC számláló, a kijelző ezeknél szeparáltan mutatja az érzékelt hibákat.

A korábbi években méréseket végeztünk a PC teljesítőképességének megállapítására, és azt találtuk, hogy 2-300 Mbps sebességnél van a működés határa. Ezeknél a vizsgálatoknál egy alkalmazás futott a PC-n és kívülről töltöttük fel az adatfolyamot idegen packetekkel.

A jitter generátort bemutató cikkünkben a Wireshark programot használtuk fel a packetek érkezési idejének rögzítésére. Ebben a programban csak a packet első 68 bájtyát (snaplen) tároltuk részben a PC terhelésének csökkentése érdekében. Nézzük meg hogyan viselkedik a PC, ha ezt a 800 Mbps-os adatfolyamot kell rögzítenie. A 3. ábra szerinti felvételen látható, hogy a packetek nagy száma túlvezérli a kijelzőt, mint azt vártuk is. A 2,5 ms-os időkapuban érkező packetek darabszáma jellemzően 180 és 200 között van. Ami furcsa, hogy a számok között nagyon ritkán, de az 1 is előfordul.



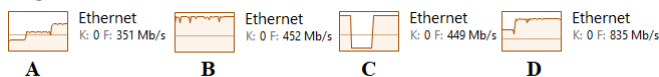
3. ábra

A PC-vel rögzített 800 Mbps-os adatfolyam

A CC hibák kiolvasására nem írtunk szoftvert, de mintavételelesen ellenőrizve a rövidke fejléceket, a CC számláló mindig 7-tel emelkedett, ami azt mutatja,

hogy nincs CC hiba a rögzített adathalmazban. Megjegyzendő, hogy csak 20 másodpercet rögzítettünk, de a PC nagyon nehezen birkózott meg a feladattal (másodpercekre elérhetetlen volt a rögzítés leállítása után).

Következő lépésben nehezítettük a PC feladatát. Első lépésben a 239.123.13.222:58222 IP címen érkező műholdas adás egyik műsorát (MPEG-2) kellett dekódolnia. A kifogástalan minőséget nyugtázva ugyanerre az IP címre küldtük az előző mérésnél használt TS Generator jelét. Az műholdvevő 38 Mbps sebességű adatfolyamához egyre nagyobb és nagyobb adatebességgel adtuk hozzá a mérő packeteket. A packeteket a 8000-es PID-en küldtük, így a műsort nem zavarták, az összegzést a switch végezte.



4. ábra

A PC bemeneti adatebessége a Win10 szerint

Mint a 4. ábrán látható, a mérőgenerátor kimeneti adatebességét 100 Mbps-os lépésekben emeltük (A). 400 Mbps-hoz érve (B) a dekódolt kép és hang még hibátlan volt, de bármilyen műveletet (az egér mozgatása, a cikk írása, rajzolás stb.) végezve a PC-vel már zavarok keletkeztek a képen. 800 Mbps-hoz érve (D) a kockásodás, elmosódás folyamatossá vált, az 5. ábrán láthatóhoz hasonló zavarok állandósultak. Közben ellenőrzés céljából néhány másodpercre kivettük a mérőjel kábelét (C), s megnyugatott, hogy valós értékeket látunk.

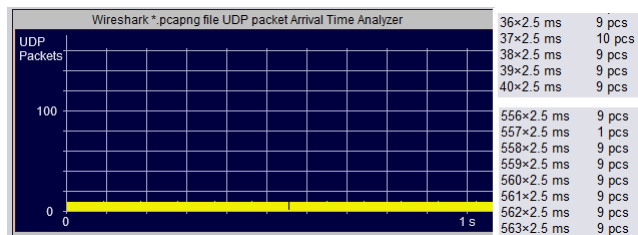


5. ábra

A dekódolt kép 835 Mb/s sebességű adatfolyamánál.

A műholdas adás adatfolyama egyenletes sebességgel érkezik. Nézzük meg a jitter generátor analízátorával, hogy milyen a packetek eloszlása a multicastos switch után, ha más adatfolyammal nem terheljük a switch-et.

Első ránézésre szép egyenletes eloszlást látunk, a 2,5 ms-os időkapukban többnyire 9 darab 7 TS packetet tartalmazó UDP érkezik. Rendszeresen egy-egy 10 packetet tartalmazó intervallum is előfordul mivel a TS órája és az időkapu órája nincs kötve egymáshoz és a számértékek sem egymásnak egész számú többszörösei.

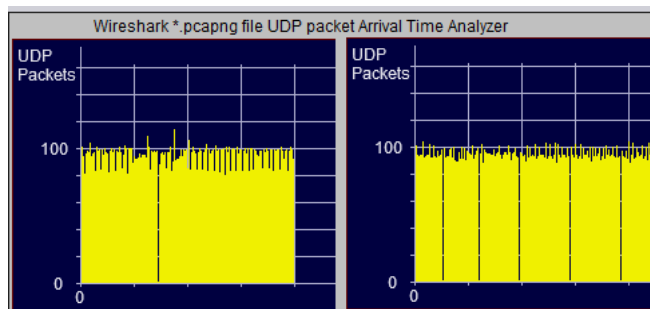


6. ábra

A műholdról érkező packetek eloszlása a switch után

A ritkán, de rendszeresen előforduló, 1 darab UDP-t tartalmazó időkapura a magyarázatot nem tudjuk, és azt sem tudjuk, hogy ez a switch vagy a PC „hibája”-e. Professzionális méréseknél (pl. a PCR mérés) mindig jusson eszünkbe, hogy itt mit tapasztaltunk.

Mivel nagyon kíváncsiak voltunk a switchek viselkedésére a következőkben azt vizsgáltuk, mi a különbség egy 5 portos mini switch és egy L2/L3 módra képes Cisco switch viselkedése között. 100, 200, 300 és 400 Mb/s sebességű adatfolyamokkal végeztünk méréseket és nagy különbséget nem találtunk, ezért csak a 400 Mb/s-os mérés eredményét mutatjuk a 7. ábrán.

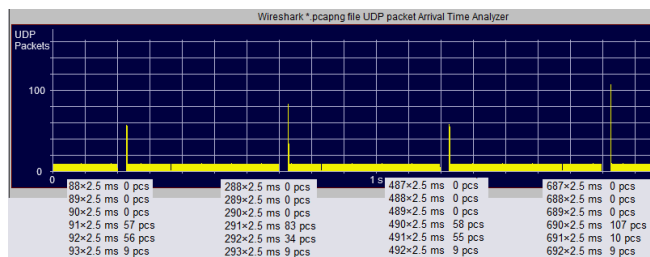


7. ábra

Bal oldalon a Cisco switch, jobb oldalon a TP-Link switch kimenetén mérhető UDP packet eloszlás 400 Mb/s esetén

A mérésből valószínű, hogy az 1 UDP/2,5 ms-os adatok a PC működéséből adódnak. Értékelésünk szerint a multicastos switch-nél a packetek időbeni egyenletlensége nem sokkal, de kicsit nagyobb.

Nehezítsük meg a switch-ek feladatát és célműszerünket használva úgy adjuk bemenetükre a műholdról érkező adatfolyamot, hogy azt jitteressé tesszük. Első lépésben gyűjtsünk össze 700 TS packetet, amely azt jelenti, hogy 100 darab UDP packetet tartunk vissza.



8. ábra

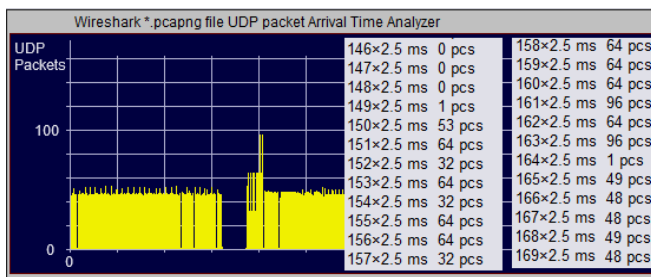
Az UDP csomagok a switch kimenetén ha ismétlődően 100-100 adatcsomagot visszatartunk majd kiengedünk

A 8. ábra szerinti mérésnél a visszatartott UDP csomagokat 10 µs-os időközökkel és nem szorosan egymást követően adtuk ki. Megnyugtató, hogy a 100-as UDP zuhatag a számokból visszaigazolható. 0 µs-os UDP távolságnál gyakori a 100-105 közötti oszlopmagasság a grafikonon.

A multicastos switch esetében 0 µs-os UDP távolságnál is egyfajta simítás tapasztalható. Néhány számsorozat diagram nélkül:

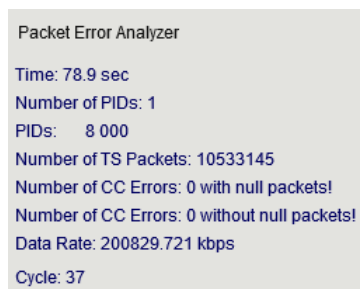
- 0, 0, 0, 28, 64, 14, 13, 9, ...
- 0, 0, 0, 51, 64, 10, 9, 9, 9, ...
- 0, 0, 0, 19, 92, 9, 9, 9, ...
- 0, 0, 0, 51, 64, 10, 9, 9, 9, ...
- 0, 0, 0, 19, 92, 9, 9, 9, ...

Az algoritmust nem ismerjük, de az látható, hogy a switch beavatkozik az UDP csomagok továbbításának időzítésébe.



9. ábra

Mivel a számolgatás hosszadalmas, megint elővesszük a mérővevőnket, s mint a 10. ábrán is látható, a packetek rendezgetése ellenére az átvitel hibátlan.



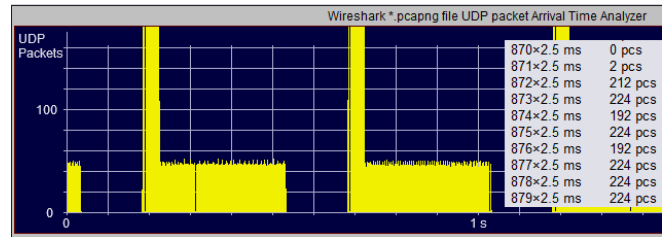
10. ábra

Hibamentes átvitel mindkét switch-nél 200 Mbps sebességű, a 9. ábra szerinti jitteres adatfolyamnál

Lépünk egy nagyot és a 200 Mbps sebességű adatfolyamba építsünk be 500 ms-os időközönként 20000 TS packetes gyűjtést. UDP-ben számolva ez 20000/7=2857 UDP visszatartását jelenti. Méretében 20000×188=3,76 MB.

Első közelítésben az átvitel mindkét switch esetében hibátlan. Grafikonunk a 11. ábra szerint alakul, de nagyon fontos kiemelni, hogy a switch semmilyen egyéb terhelést nem kap, további kábeleket nem csatlakoztattunk a portokhoz.

Csatlakoztassuk a korábban használt műholdvevő 38 Mbps sebességű adatfolyamát a switch egy szabad portjához, majd kérjük ki azt egy készülékkel.



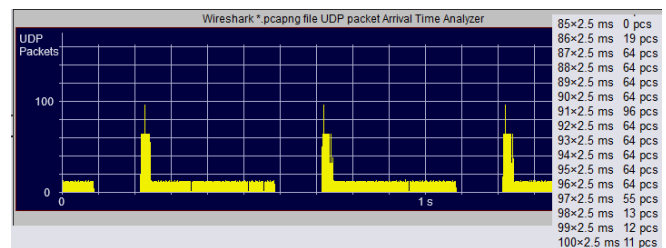
11. ábra

A switch terhelése erősen jitteres adatfolyammal

A kis méretű switch esetében a CC hibák száma rohamosan növekedni kezd. Többféle variációt kipróbálva (pl. csak bemegy a stream, de nem kérjük ki, kisebb sebességű streamet fűzünk át rajta stb.) a hibák mindig jelentkeznek, a kiegészítő adatfolyamot levéve a hibák megszűnnek.

Nem teszünk egyebet, mint a kábeleket átdugjuk a multicastos Cisco switch-be és egyetlen hibát sem találunk a variációkban.

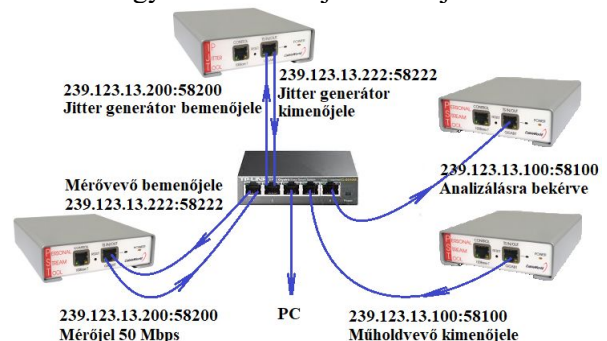
Bizonyára az olvasót is érdekli, hogy hol lehet a hiba határa, ezért egy 50 Mbit/s jitteres adatfolyam mellé tesszük a műholdas jelet és a jitter generátoron állítgatjuk a visszatartandó packetek darabszámát. Ezzel a mérési módszerrel a hibahatár 4200 TS packetes gyűjtésnél adódik. 4300-nál már egy-két hiba jelentkezik másodpercenként. A mérőjel a 12. ábrán látható.



12. ábra

A hibahatárhoz tartozó mérőjel kis switch esetén

Célszerűnek láttuk, ez utóbbi mérési összeállítás blokkvázlatát úgy megrajzolni, hogy az adatfolyamok is láthatók legyenek. Ezt a rajzot mutatja a 13. ábra.



13. ábra

A 12. ábra szerint mérés blokkvázlata

Hol van ezen eszközök teljesítőképességének a határa? A választ nem tudjuk, de felhívjuk a figyelmet arra, hogy jitteres adatfolyamoknál nagyon gondosan kell a switch típusát megválasztani.

Jitter-mentesítés

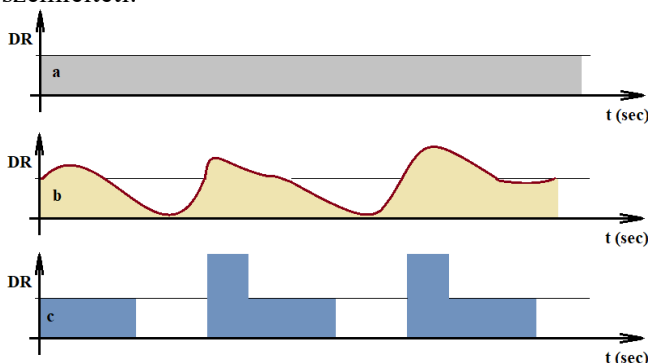
Az IP Stream Tool bemutatása

Az IP hálózaton fellépő jiterről akkor beszélünk, ha a hálózat kimeneti pontján az adatcsomagok áramlása, pontosabban időbeni eloszlása nem egyezik meg a betáplálási pont áramlási sebességével. A konstans időbeli eltolás nem számít jitternek, és ugyanígy most nem tekintjük jitternek néhány (1, 2 vagy 10) adatcsomag időbeli elcsúszását.

A jitter többségében a nagyméretű távközlési hálózatokon keletkezik, jellemzően ott, ahol a rádió- és televízió műsorok adatfolyamai az internet szolgáltatások adatcsomagjaival közösen kerülnek továbbításra.

A beszéd típusú szolgáltatások (telefon, skype és hasonló) esetében természetesnek vesszük, hogy az adatcsomagok nem késleltethetők, mivel a felhasználók azonnal észreveszik, ha a hang akadozik, vagy nagyobb mértékben késik. Cikkünkben egy olyan készüléket mutatunk be, amelyik a rádió- és televízió műsorok adatfolyamai esetében segít az eredeti adatfolyam, a jittertől mentes adatfolyam visszaállításában.

Előző számunkban egy olyan célműszert (Jitter Generator) mutattunk be, amelynek segítségével a szimulálni tudjuk az IP hálózaton keletkező jittert. A valóságos jitter előre nem meghatározható, szabálytalan adatcsomag eltolódások formájában jelentkezik, amelyet legjobban a pillanatnyi adatsebességgel lehet jellemezni. Az 1. ábra felső részén egy egyenes sebességgel áramló adatfolyamot láthatunk. A középső ábra azt szemlélteti, milyen lesz az adatfolyam időbeli lefolyása, ha az átvitel során jitter keletkezik. A harmadik görbe a célműszerünkkel előállítható jittert szemlélteti.



1. ábra

- a) jitter mentes adatfolyam
- b) jitteres adatfolyam a valóságban
- c) a célműszerünkkel előállítható jitteres adatfolyam

A jitter-mentesítésre szolgáló készülék neve: IP Jitter Tool, fényképe a 2. ábrán látható.



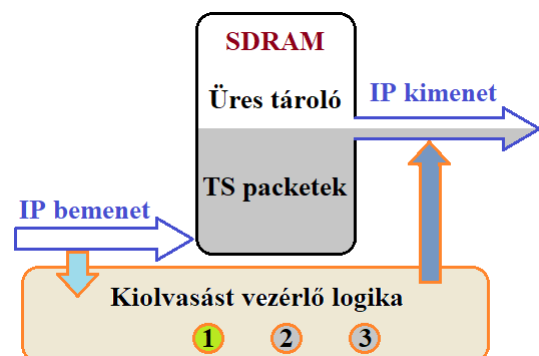
2. ábra

Az IP Jitter Tool fényképe

Mielőtt bárki reklamálna, jelezzük, hogy a jitter generátornak is IP Stream Tool a neve és csak a típuszámból lehet eldönteni, melyik van a dobozban. A jitter generátor célműszernek készült ezért nem csináltattunk neki saját előlapot.

A jitter mentesítő működési elve nagyon egyszerűnek látszik. A készülék SDRAM-ba írja a bemenetére érkező TS packeteket, majd egy, a felhasználó által konfigurált algoritmus szerint igyekszik elvárásainknak megfelelően kiadni a packeteket. Mint látni fogjuk az algoritmusok megírása meglehetősen bonyolult, de a felhasználó által történő konfiguráció sem egyszerű. A nehézség abból adódik, hogy minden hálózaton más a jitter jellege, és a felhasználónak a saját jitteréhez igazodva kell megtalálnia a helyes eljárást.

A részletes ismertető előtt elmondjuk még, hogy a készülék két csatornás, azaz egyidejűleg két adatfolyam mentesítésére képes. Azt hogy a készülék két csatornás lett, a hardver adottságai határozták meg, ugyanis ennek a készüléknek is az alapját a Gigabit Ethernet Controller II. panelünk képezi, mindössze a betöltött programok újak. A jitter mentesítő működési elvét a 3. ábra szemlélteti.



3. ábra

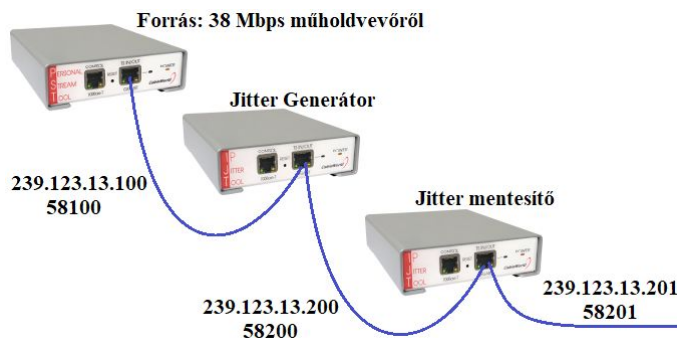
A jitter-mentesítő működési elvének vázlata

Az SDRAM mérete 32 MB. A vezérlő logika a következő négy üzemmód egyikére állítható:

- Az adatfolyam átengedése (Bypass Mode)
- Állandó kimeneti adatsebesség tartás
- RTP pozíció visszaállítás
- PCR visszaállítás

1. Az adatfolyam átengedése

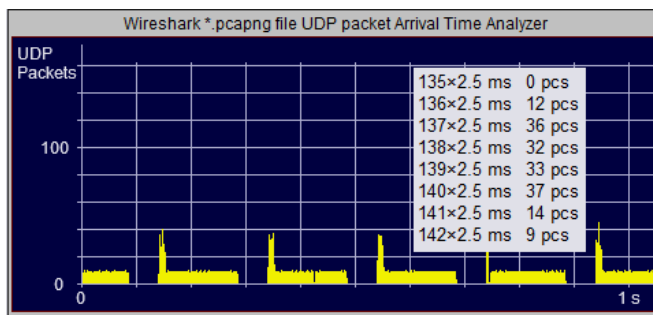
Az adatfolyam módosítás nélküli átengedése csak mérésekhez és ellenőrzésekhez kell. Ebben az üzemmódban a készülék változtatás, azaz időbeli eltolás nélkül adja ki a bemenetére érkező packeteket. Első tesztünkhöz mérőjelként használjuk fel a műholdvevő kimenőjelét, amelyik a 239.123.13.100:58100 IP címen jelenik meg a vevő kimenetén. A jitter generátoron áthaladva a 239.123.13.200:58200 IP címen található az adatfolyam. A jitter mentesítő bekérve ezt az adatfolyamot feldolgozás után a 239.123.13.201:58201 címen adja ki azt. Mérési összeállításunkat az 4. ábra szemlélteti.



4. ábra

A mérési összeállítás elvi rajza (a switch, a mérőműszer és a PC nélkül)

Jitter generátorunkat állítsuk be úgy, hogy 200 ms-os ismétléssel 1400 TS packetet, azaz 200 UDP-t tartson vissza, majd a kimenetén 10µs-s időközökkel adja ki azokat. Switchként a multicastos Cisco switch-et használjuk. A 5. ábrán már úgy mutatjuk a mérőjelet, hogy a bufferen is áthaladt, de az bypass módra volt állítva.



5. ábra

A vizsgálatainkhoz használt mérőjel

A mérőjelet az SW-4811 szoftverünkkel hívtuk be a PC-be és egyúttal megvizsgáltuk a dekódolt képet is.

A dekódolt kép és a hang olyan érdekes volt, hogy azt célszerűnek tartjuk olvasóinknak is megmutatni. A kép nem fagyott le, a mozgás folyamatos volt, viszont a gyorsan változó képrészletek erősen kockásodtak. A 6. ábra egy jellemző képrészletet mutat.



6. ábra

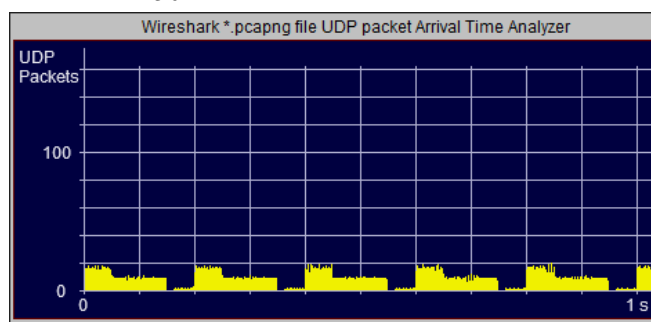
Az 5. ábra szerinti jel dekódolás után

A kép mellett a hang nehezen érthető, ritmikusan lüktető, kellemetlen hang volt.

2. Állandó kimeneti adatsebesség tartás

A sebesség tartó logika egy idő intervallum beállítását várja a felhasználotól. A készülék a beállított intervallum alatt SDRAM-ba írja a packeteket és az intervallum végén kiszámítja az átlagsebességet. A következő intervallumban ezzel a sebességgel adja ki az SDRAM tartalmát, miközben gyűjti a bemenetére érkező packeteket és ciklikusan ismétli a folyamatot.

A szemléltetés érdekében elsőként szándékosan egy rossz beállítást mutatunk (7. ábra), mivel jó beállítás esetén nincs mit mutatni. Ebben a beállításban az idő intervallum 50 ms.



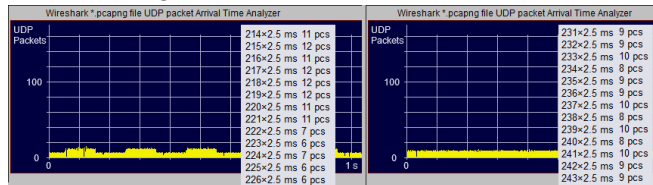
7. ábra

A jitter simítása 50 ms intervallum beállításával

A grafikon azt mutatja, hogy történt ugyan valami, de valójában ennél többet várunk a készüléktől. Ezzel szemben a dekódolt kép jelentős mértékben javult, a hangban alig lehet itt-ott egy-egy hibát észrevenni.

Lépünk egyet előre és az intervallum értékét állítsuk 100 majd 200 ms-ra. A 8. ábra mutatja, hogy a 100 ms-os ablak választása jobb megoldás, de még nem tökéletes.

Mint látható a 200 ms-os ablak választása akár tökéletes megoldásnak is nevezhető.



8. ábra

Simitás 100 ms-os (bal oldali görbe) és 200 ms-os ablakkal

Mielőtt elégedetten hátradölnénk székünkben, nézzük meg a megoldás előnyeit és hátrányait.

1. A fejlesztés során azt tapasztaltuk, hogy ez egy igen stabil és jól működő megoldás. Minden alkalmazásban elsőként ennek beállítását javasoljuk.
2. Állandó sebességű, CBR streamek esetében nagy idő ablakkal dolgozva jó kimeneti streamet kapunk, de a teljes stream pontosan annyi késleltetéssel jelenik meg a kimeneten, mint amennyi az idő ablak nagysága.
3. VBR streamek esetén is alkalmazható, különösen akkor, ha nem túlzottan gyors az adatfolyam időbeli változása. Ilyenkor a kisebb idő ablak alkalmazása látszik jobbnak, ha ezt a jitter jellege megengedi.

Megjegyzendő, hogy a készülék csatornánként egy-egy 32 MB-os, azaz $32 \times 8 = 256$ Mbit-es SDRAM-mal dolgozik, amelyeket nem illik csurig tölteni. Durván számolva, a 38 Mbps sebességű adatfolyamnál 6 másodpercnyi adatfolyam fér a tárbá. A jitter jellegéből lehet eldönteni, hogy meddig célszerű tölteni a tárolót azért, hogy egyetlen ciklusban se legyen túlcserélődés.

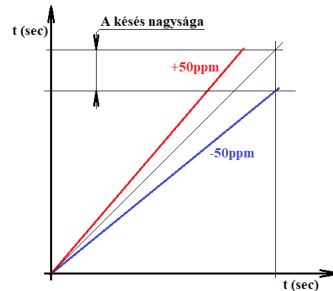
3. Szabályozás az RTP időbélyegére

Az RTP (Real Time Protocol) az UDP formátum néhány bajttal kiegészített változata. A kiegészítés többek között tartalmaz egy 16 bites sequence numbert, amelynek pontosan ugyanaz a feladata, mint a packetek továbbításánál használt 4 bites CC-nek (Continuity Counter). A másik, számunkra érdekes kiegészítés 32 bites időbélyeg (Time Stamp), amely mutatja számunkra a feladás időpontját. Az időbélyeg egy folyamatosan növekvő szám, de az alapfrekvencia alkalmazásfüggő.

Mondhatnánk, hogy nincs egyszerűbb feladat, mint a jitteres IP hálózatról beérkező adatsomagokat kicsit tárolni, majd az időbélyegnek megfelelő időpontban kiadni. Nézzük meg mennyiben igaz ez az állítás.

A készülékekben használt oszcillátorok pontossága jellemzően 20...50 ppm, ami azt jelenti, hogy a tényleges frekvencia értéke kisebb, mint $20...50 \times 10^{-6}$ értékkel tér el a névleges értéktől. Közben megjegyezzük, hogy az 1...2 ppm csak különleges és drága eszközökkel érhető el.

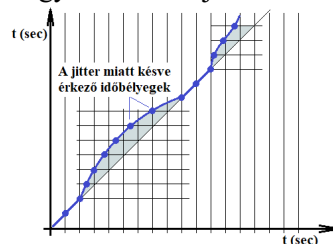
Feltételezve, hogy az adó oldal abszolút pontos oszcillátorral dolgozik, a 9. ábra szerinti esetek állhatnak elő a vételi oldalon 50 ppm-es oszcillátort használva. Amikor lassabban jár az oszcillátorunk, kb. 5,5 óra múlva lesz 1 másodperc nagyságú a lemaradásunk.



9. ábra
A ± 50 ppm-es eltérés szemléltetése

Az adó oldali oszcillátor azonos, de ellenkező előjelű hibája esetén az eltérés fele ennyi idő alatt áll be, és így tovább. E néhány változat is jól szemlélteti, hogy valamilyen szabályozás, az órák összehangolása nélkül nincs lehetőség a helyes működés fenntartására.

A 10. ábrán látható, hogy a jitter hatására a beérkező időbélyegeket vagy pontosak (nincs jitter) vagy késnek a jitter miatt. Mivel nem tudjuk, hogy melyik időbélyeg pontos és melyik mekkora késéssel érkezett, itt az átlagszámításnál bonyolultabb szabályozásra van szükség.



10. ábra A jitter hatása az időbélyegekre

Amikor a mi óránk lassabban jár, az SDRAM tartalma növekedést mutat, és fordítva. Ehhez hasonlóan más jellemzők figyelése is bevonható a szabályozásba. A helyi órát szabályozó eljárás dönti el, hogy az adott készülék milyen jellegű és milyen mértékű jitter megszüntetésére alkalmas.

A CableWorld megoldásában két szabályozási rendszer kifejlesztését tervezzük. Az egyik az RTP időbélyegre történő szabályozás, a másik egy PCR visszaállítására törekvő szabályozás. Nagyon fontos, hogy ez utóbbinál nem a PCR korrigálása történik, ugyanis feltételezzük, hogy a küldő oldalon a PCR még hibátlan volt. A szabályozó áramkör ezt az állapotot igyekszik visszaállítani.

Az RTP időbélyegére történő szabályozás előnye, hogy a felhasználónak nem kell különböző paramétereket beállítania. Hátránya, hogy csak RTP formátumú adatsomagoknál használható, és ezekhez is csak akkor, ha az időbélyeg a 90 kHz-es órajelből került kialakításra.

A PCR visszaállítására törekvő szabályozás hátránya, hogy a felhasználónak egy PID értéket is meg kell adnia. A jelenlegi fázisban az RTP-re történő szabályozást már valós rendszerekben teszteljük, a PCR-es változat fejlesztése befejezés előtt áll.

DVB T2-MI

Korábbi cikkeinkben már említettük, hogy a DVB-T2 modulátor lehetővé teszi egymástól független adatfolyamok más-más modulációs paraméterekkel történő kisugárzását. Ezek a független adatfolyamok un. PLP (Physical Layer Pipe) formába építve érkeznek a modulátorhoz.

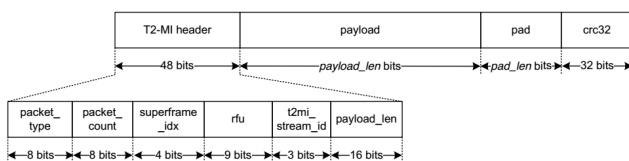
A szabvány rögzíti, hogy a modulátor előtti adatfeldolgozás, a PLP-k kialakítása nem része a modulátornak. A modulátor előtti adatfeldolgozó egység neve „T2-gateway”, a modulátor innen a T2-MI interfészen keresztül kapja az előfeldolgozott moduláló jelet.

A T2-MI adatfolyam kialakításának és továbbításának részletei az ETSI TS 102 733 szabványban találhatók. Cikkünkben e témakörrel kívánunk könnyen értelmezhető képet adni.

A T2-MI adatfolyamot egyidejűleg több modulátorhoz is el kell juttatni (pl. országos hálózatok, vagy SFN adók esetében), ezért az adatfolyamot úgy alakították ki, hogy IP-n és ASI vonalon, vagy akár műholdon keresztül is továbbítható legyen. Előre jelezzük, hogy a PLP-k (Physical Layer Pipe) adatfolyamának szerkezete jelentősen eltér a Transport Stream packetek szerkezetétől, ezért a TS packetekbe történő csomagolást „kényszerzubbonynak” kell tekinteni.

A T2-MI csomagokat tehát a már jól ismert TS packetek továbbítják, és a PAT, illetve PMT tábla mutatja az utat a packetekből történő összefűzéshez. A PMT tábla által mutatott PID-en (pl. 6-os típusú stream) a Payload Unit Start Indicator-tól kezdve hosszú adatsomagokat kell összerakni, különösen ügyelve a pointer-re és a takarékos összefűzésre.

A kibontott T2-MI adatfolyam szintén csomagokból áll (T2-MI packet), amelyek fejlécből és payload-ból állnak (1. ábra). A fejlécben a payload adattartalmáról kapunk részletes információt, többek között a packet típusáról (Baseband frame, Auxiliary stream I/Q data, L1 current, L1 future, stb), a packet sorszámáról és egyéb jellemzőiről. Itt van megadva továbbá a payload hossza is. A T2-MI packet végén 32 bites CRC kódot találunk.



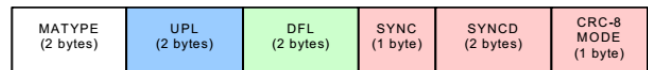
1. ábra

A T2-MI packet (ETSI TS 102 773)

A kép és hang adatok a T2-MI csomagok által szállított Baseband frame (BBFrame) keretekben vannak. Ezek részletes felépítése a DVB-T2

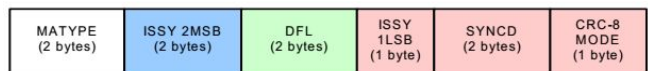
szabványban (ETSI EN 302 755) található. (A T2 szabvány az S2-höz hasonlóan sokféle, fix vagy változó hosszúságú csomagokból álló, vagy folytonos bemeneti stream formátum feldolgozására alkalmas, de mi most csak a műsorszórásban használatos MPEG-2 Transport Stream (TS) formátummal foglalkozunk.)

A BBFrame szintén fejlécből és payload-ból áll, ez utóbbit most adatmezőnek (data field) hívják és hossza (DFL) bitekben számolandó. A 80 bitből (10 bájtból) álló fejléc az üzemmódtól függően kétféle felépítésű lehet: Normal (NM), vagy High Efficiency (HEM). A 2. ábra mutatja az NM, a 3. ábra pedig a HEM formátumú fejlécet.



2. ábra

NM BB fejléc formátum (ETSI EN 302 755)



3. ábra

HEM BB fejléc formátum (ETSI EN 302 755)

Mindkét formátum esetében az első két bájt (MATYPE) felépítése azonos.

A MATYPE első bájtjának összetétele:

- ◆ TS/GS: (2 bit) mutatja a stream formátumát, TS esetén értéke bin11
- ◆ SIS/MIS (1 bit): egyetlen vagy többszörös bemeneti stream kerül-e átvitelre. Egyetlen bemeneti stream esetén 1.
- ◆ CCM/ACM (1 bit): konstans kódolást és modulációt vagy változó paraméterű kódolást és modulációt alkalmaztak-e. CCM: 1
- ◆ ISSYI (1 bit): ha az Input Stream Synchronization Indicator bit értéke egy, a bemeneti stream szinkronizációs mechanizmus aktív, az ISSY mező (lásd később) értéke felhasználandó a szinkronizáláshoz.
- ◆ NPD (1 bit): ha értéke egy, a Null packet törlési mechanizmus aktív, a Null packetek törlésre kerülnek.
- ◆ EXT (2 bit): a maradék két bit jelenleg nincs használva.

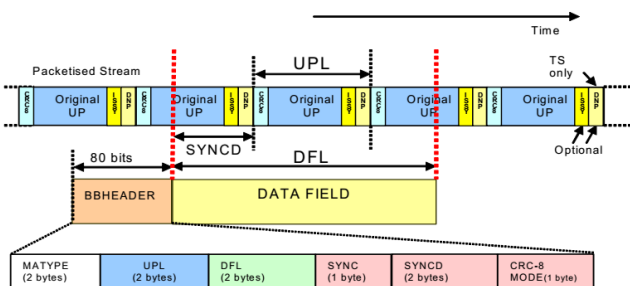
A MATYPE második bájtjának jelentése:

Ha a SYS/MIS változó értéke nulla (többszörös bemeneti stream), ez a bájt tartalmazza az adott bemeneti stream azonosítóját (ISI). Fontos tudni, hogy az ISI a DVB-S2 rendszerből került a DVB-T2 szabványba, és jelentése azonos az itt használt PLP_ID paraméter jelentésével.

Az NM formátum és csomagokból álló bemeneti stream esetén a fejléc további paramétereinek értelmezése:

- ◆ Az opcionális ISSY (Input Stream Synchronization) mechanizmus használata esetén az UP (user packet) végéhez 16 vagy 24 bites ISSY mezőt illesztnek és ennek megfelelően az UPL (user packet hossz) értéke is 16 vagy 24 bittel nő. A modulátor órajelével meghajtott számláló aktuális értékét tartalmazzák az ISSY bájtok, és arra szolgálnak, hogy a vevőkészülék képes legyen a kimeneti adatfolyamot helyes időzítéssel helyreállítani. Ha az ISSY mechanizmus nincs használva, a három bájt értéke nulla.
- ◆ A UP elején lévő szinkron bájt törlésre kerül, az UPL értéke 8 bittel csökken, és a szinkron bájt a fejléc SYNC mezőjében kerül tárolásra.
- ◆ Az opcionális null packet törlési mechanizmus használata esetén a törölt null packetek száma a DNP bájtba kerül és a következő továbbított packethez csatolják. Az UPL értéke 8 bittel nő.
- ◆ CRC-8 ellenőrző kód található a UP legelején, a szinkron bájt helyén, amely az előző packet adatai alapján kerül kiszámításra, tehát az előző UP hibamentességét lehet ellenőrizni vele.
- ◆ A SYNCD mező az adatmezőben továbbított user packetek feldolgozását segíti. Értéke az adatmezőben továbbított első teljes UP első bitjének távolságát adja meg az adatmező kezdetétől számítva. A UP első bitjét jelen esetben (az előző fejléchez tartozó) CRC-8 bájt első bitje jelenti. A UP vége a DNP vagy ISSY mező, amennyiben ezek jelen vannak.
- ◆ A fejléc utolsó bájtja egy CRC-8 kód, amely a fejléc előző kilenc bájtjára kerül kiszámításra, tehát a fejléc bájtjainak hibamentessége ellenőrizhető vele. Ezen túlmenően, ha a számított értéket ezzel a tizedik bájttal EXOR kapcsolatba hozzuk, az eredmény jelzi, hogy az adott fejléc milyen kialakítású. Nulla eredmény esetén a fejléc NM formátumú, az egy érték pedig a HEM formátumot jelzi.

A BBFrame felépítését benne a UP packetekkel NM formátumú fejléc esetén a 4. ábra mutatja.



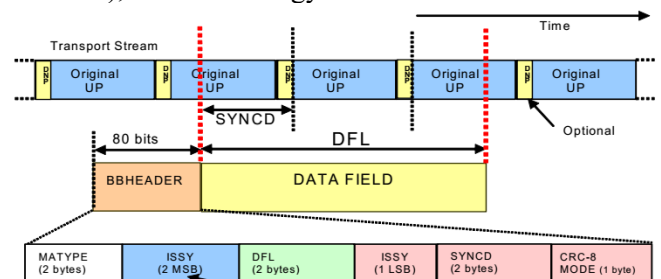
4. ábra

BBFrame felépítése, normál mód (NM)(ETSI EN 302 755)

A mi esetünkben, amikor a BBFrame adatmezője TS packeteket tartalmaz, ismert a UP packetek hossza (UPL = 188x8 bit), valamint a szinkron bájt (SYNC = hex47) is. Ezért ez a három bájt más célra (az ISSY átvitelére) használható. Ezért TS packetek átvitelkor leggyakrabban a HEM formátumot használják.

A HEM formátum esetén a fejléc jellemzői:

- ◆ Az ISSY bájtjai az UPL és a SYNC mezők helyére kerülnek.
- ◆ Az eredeti szinkron bájt (hex47) semmilyen formában nem kerül továbbításra.
- ◆ Az opcionális null packet törlési eljárás (DNP) megegyezik az NM esetén használttal.
- ◆ UP szinten CRC-8 számítás és továbbítás nem történik. (Természetesen a fejlécre vonatkozó CRC-8 ugyanúgy jelen van, mint az NM fejléc esetén. Lásd 4. és 5. ábra.)
- ◆ A SYNCD mező az adatmezőben továbbított user packetek feldolgozását segíti. Értéke az adatmezőben továbbított első teljes UP első bitjének távolságát adja meg az adatmező kezdetétől számítva. A UP első bitje az eredeti UP első bitjének felel meg a szinkron bájt eltávolítása után. A UP vége a DNP mező utolsó bitje, amennyiben jelen van a DNP.
- ◆ Az UPL nem kerül továbbításra, hiszen értéke ismert: 188x8 -8 (szinkron bájt) +8 (opcionális DNP), tehát 1496 vagy 1504 bit.



5. ábra

BBFrame felépítése, high efficiency mód (HEM) (ETSI EN 302 755)

A fentiek ismeretében és egy T2-MI stream minta birtokában kísérletképpen megpróbálkozhatunk a T2-MI packetekből visszanyerni a továbbított (TV és rádió programokat tartalmazó) transport streamet vagy streameket.

Ahogy már korábban volt szó róla, a T2-MI stream-et is TS packetek hordozzák, tehát a T2-MI csomagok összeállítása a megfelelő TS packetek feldolgozásával történik. A PMT táblában mindig 6-os típusú streamként szerepel a T2-MI stream, tehát ezeket a packeteket kell kigyűjtenünk a megfelelő PID ismeretében.

A PMT táblában T2-MI descriptor is előfordulhat, amely a descriptorok nagy száma miatt ún. extension típusú descriptor, aminek jellemzője, hogy van egy

közös címkéje (descriptor tag: hex7F), és egy egyedi címkéje (descriptor tag extension: hex11). A descriptor tartalmaz egy t2mi_stream_id adatot, ami az egyidejűleg továbbított T2-MI streamek egyedi azonosítására szolgál, valamint egy num_t2mi_streams_minus_one értéket, ami megadja, hogy hány T2-MI stream szükséges a teljes DVB-T2 stream előállításához.

Ha a T2-MI streamet alkotó TS packetek összegyűjtése megvan, első lépésben egy olyan TS packetet kell megkeresnünk, amelyikben a payload_unit_start_indicator értéke egy, ugyanis ebben kezdődik a T2-MI adatfolyamának valamelyik adategysége (ha szerencsénk van, akkor éppen egy BBFrame).

Ezután vizsgáljuk meg az adaptation_field_control bitek segítségével, hogy az adott TS packet fejléce tartalmaz-e adaptációs mezőt, hiszen ebben az esetben a fejléc hossza az adaptációs mező hosszával (adaptation_field_length) megnő. Ha ennek figyelembe vételével eljutottunk a fejléc végére, a payload első bájtja most a pointer_field.

A pointer által megadott számú bájtot átugorva, a packet végéig megmaradt bájtokat és a további packetekben szállított 184 bájtokat összefűzve (a fejléceket közben átugorva) egy hosszú bájt sorozatot kapunk.

A következő T2-MI packet kezdetét egy olyan TS packet jelzi, amelyben a payload_unit_start_indicator újra egy értékű.

A T2-MI packetek összeállítása után megkezdhetjük ezek feldolgozását. A packet 48 bites fejléccel kezdődik, amelynek első bájtja a packet_type. Ez mutatja meg a packet által szállított DVB-T2 szabvány szerinti adattartalom fajtáját.

A T2 modulátor, illetve a vevőkészülék hibátlan működéséhez sokféle frame és adat átvitele szükséges, számunkra most csak a már korábban ismertett BBFrame érdekes, amelynek azonosítója (packet_type) hex00. A többi packet típust most figyelmen kívül hagyjuk.

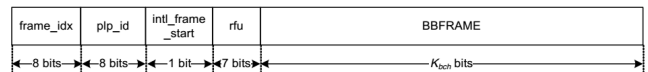
A fejléc információi közül még a payload_len fontos számunkra, amely a packet hosszát jelzi bitekben. Az általam vizsgált mintában ez 50000 ... 60000 bit közé esett. A pad_len bit azt mutatja meg, hogy esetlegesen hány bitet kell még hozzáadni a packet végéhez, hogy bájt határon végződjön, tehát a packet nyolccal osztható legyen. Ilyenkor a kitöltő bitek 0 értékűek. A mintában én ilyet nem találtam. A T2-MI packet legvégén négy bájtos CRC32 ellenőrző kódot találunk, amit a packet teljes adattartalmára kell számolni (header + payload) az esetleges kitöltő bitekkel együtt.

A megfelelő T2-MI packetek (packet_type = hex00) kiszűrése és a fejléc feldolgozása után nézzük, hogy mit találunk a payload-ban (6. ábra).

Az első bájt a frame_idx, amely az aktuális T2-frame sorszama a super-frame-en belül (ETSI EN 302 755). Nekünk ezzel most nincs dolgunk.

Ahogy korábban már volt róla szó, a DVB-T2 rendszerben egyidejűleg több transport stream átvitelére is van lehetőség. Azonos TS-hez tartozó adatokat hordozó adatfolyamot nevezik a DVB-T2 rendszerben PLP-nek (Physical Layer Pipe). Azt, hogy az adott BBFrame melyik TS adatait tartalmazza, a plp_id adja meg. A payload második bájtjában találjuk ezt az azonosítót. Jelen példánkban válasszuk a plp_id = 0 értéket, így ha kiszűrjük a nulla azonosítójú BBFrame-eket, abból összeállítható egy hibátlan, koherens TS.

A harmadik bájtban csak a legfelső bite hordoz hasznos információt. Értéke egy, ha az adott PLP esetén a packet az első BBFrame-t tartalmazza a közbeszövött keretek között, nulla, ha a további kereteket. Esetünkben nincs rá szükségünk.



6. ábra

T2-MI Baseband Frame payload

Ezzel eljutottunk a BBFrame kezdetéig. Fontos tudni, hogy egy BBFrame mindig egyetlen T2-MI packetbe van becsomagolva, egyéb kitöltő bájtok nélkül. A BBHeader feldolgozását a cikk elején megismertek szerint tudjuk elvégezni. Számoljuk ki a CRC-8 kódot és hasonlítsuk össze a 10 bájtos fejléc utolsó bájtjával. Képezzük a két érték EXOR kapcsolatát, ebből megtudjuk, hogy a frame NM vagy HEM formátumú-e. Ha az eredmény egy, a frame HEM formátumú, nulla esetén pedig NM. Az általam vizsgált mintában a HEM üzemmódot használták, tehát a továbbiak erre vonatkoznak.

A fejléc feldolgozása után már ismerjük a BBFrame adatmezejének hosszát (DFL), valamint a SYNCN értékét és azt, hogy a NULL packet törlési eljárást jelen esetben nem alkalmazták (NPD=0). Ezek alapján elkezdhetjük a TS packetek kigyűjtését a frame-ből és ezzel az eredeti TS összeállítását. Fontos, hogy mivel a TS packet szinkron bájtja (hex47) nem kerül átvitelre, ezt minden TS packet kezdetén pótolnunk kell. Ezután a SYNCN ponttól kezdve 187 darab bájtot kimásolunk és ezzel készen is van az első TS packetünk. Jelen mintában tehát nem történt NULL packet törlés, ellenkező esetben az elkészült packet elé DNP darab szabványos Null packetet (PID = 0x1FFF) kellene beilleszteni. A második packethez ismét beillesztünk egy szinkron bájtot, majd 187 adatbájtot másolunk mögé. Ezt addig ismételjük, amíg a DFL-SYNCN hosszú mező végére nem érünk.

Nagyon ritkán fordul elő, hogy a BBFrame adatmezejének vége egybeesik az utolsó TS packet végével. Általában a TS packet egy része már nem fér

bele az adatmezőbe. Ilyenkor a rész packet-et bemásoljuk az adatmező végéig (nem megfelelkezve elsőként a szinkron bájtt beillesztéséről), majd a 188-ig még hiányzó bájtokat a következő BBFrame adatmezejének elejéről a packetünk végére másoljuk. Az új frame fejlécének adataival a SYNCD ponttól a fentiek alapján folytatjuk a TS packetek összeállítását míg az összes frame-t fel nem dolgoztuk a mintából. Összefoglalva a folyamatot:

1. A megfelelő TS packetekből összeállítottuk a T2-MI packeteket.
2. A T2-MI fejléc segítségével kiválogattuk a T2-MI packetekből a BBFrame típusú payload-ot hordozókat, és ezek közül is a kívánt PLP azonosítójukat.
3. A BBFrame payload-ból kigyűjtöttük és helyreállítottuk az eredeti TS packeteket. Ezzel visszaállítottuk a DVB-T2 modulátorok számára előkészített adatokból a kisugárzásra szánt TS-t, benne a TV és rádió programokkal.

Personal Stream Tool

Extra

A Personal Stream Tool működésének ismertetése során már többször beszéltünk arról, hogy a készülékben van egy Real Time Analyzer modul. Többben is kérdéseket tettek fel ennek felépítésével és működésével kapcsolatban, ezért a következőkben ezt a modult mutatjuk be

Elsőként nagyon fontos látni, hogy a PST FPGA áramköre úgy került megtervezésre, hogy a gigabiten érkező packetek mindegyikét fel tudja dolgozni. Mivel a TS packet $188 \times 8 = 1504$ bitből áll, így a TS packetek feldolgozására $1504 \times 1 \text{ ns} = 1,504 \mu\text{s}$ áll rendelkezésünkre. Miután az IP szűrők eldöntötték, hogy a packetet beengedik, megkezdődik annak feldolgozása.

A Real Time Analyzer az SDRAM egy szeparált egységében tárolja az információkat. A modul elsőként a PID értékét olvassa ki és eggyel növeli az adott PID-hez tartozó darabszám értékét. A PID érték mellett a CC számláló állapota is rögzítésre kerül a PID első megjelenésénél. A további esetekben a CC hibafelismerő dolgozza fel a CC értékét, ügyelve arra, hogy az Adaptation Field egyes változatainál nem kell lépnie a CC számlálónak, illetve tárolja a beérkezett CC értéket. Hiba esetén eggyel növeli a hibák számlálóját.

A Real Time Analyzer modul egy-egy flag-et billent be, ha kódolt jelzést vagy TEI hibajelzést lát a packetban, illetve PCR adatot talál. A fejlesztés során ezek mellett egyéb információkat is rögzítettünk a packetekről, de ezekről ráérünk majd később említést tenni. Alapesetben a 64 csatornához tartozó SDRAM-ot egyetlen utasítással lehet törölni, a törlési idő 32 ms. A törlést követően csak nagyon ritkán (akár több nap vagy hét után) van szükség újabb törlésre, ha a transport streamekben kevés hiba van.

A PID szerinti darabszám számlálója 4 bájtos, a CC hibák számlálója 2 bájtos. E két adatból pontosan kiszámítható, hogy hány hiba után, vagy adott sebesség esetén hány TS packet után van szükség törlésre, ahhoz, hogy egyetlen adatunk se vesszen el.

Az SDRAM tartalmának kiolvasása hosszadalmas lehet ugyanis valamennyi PID értékhez rendeltünk egy-egy 16 bájtos tárolót. Az olvasás gyorsítására írtunk egy TCP-s kiolvasó eljárást, amelyben a belső mikrokontroller elvégzi az adatok előzetes kiértékelését, és csak azon PID értékek tárolójának tartalmát küldi ki, amelyeknél legalább egy TS packet érkezett. A válasz tartalmazza a törlés óta eltelt időt, majd a PID érték két bájtos változója után a 16 bájtos információs blokkot.

Mint látható, jól működő rendszereknél többnyire csak olvasni és kijelezni kell az adatokat. Ezzel szemben zavarok (pl. villámcsapás hatására bekövetkező stream hibák stb.) esetén célszerű törölni az információs blokkot és a hiba megszűnte után tiszta lappal folytatni a megfigyelést.

Törlés esetén a törlési idő alatt az SDRAM nem tud adatokat fogadni, ezért erre az időre egy segéd tárolóba írja a modul az adatokat, azonban ilyenkor a feldolgozó számítógépnek kell tárolnia a korábbi, a törlés előtti adatokat, majd ehhez hozzáadni az új ciklus adatait. Nagyobb mérőrendszereknél felvetődött, hogy jobb lenne, ha csatornánként is lehetne törölni az információs blokkot, ezért a v1.24-es változatba beépítettük a csatornánkénti törlés lehetőségét. A csatorna adatainak törlésekor is az előzmények tárolása a kiolvasó számítógép feladata. Egy csatorna adatainak törlési ideje 500 μs .