Cross-layer forgalom- és protokoll elemzés

Cross-layer network traffic and protocol analysis

Doktori (PhD) értekezés

OROSZ PÉTER

Témavezető: Prof. Sztrik János

Debreceni Egyetem
Természettudományi Doktori Tanács
Informatikai Tudományok Doktori Iskola
Debrecen, 2009.

Ezen értekezést a Debreceni Egyetem Természettudományi Doktori Tanács Informatikai Tudományok Doktori Iskola Informatikai Rendszerek és hálózatok programja keretében készítettem a Debreceni Egyetem természettudományi/műszaki doktori (PhD) fokozatának elnyerése céljából.

Debrecen, 2009……

……..……
Orosz Péter
jelölt

Tanúsítom, hogy Orosz Péter doktorjelölt 2005-2009 között a fent megnevezett Doktori Iskola Informatikai rendszerek és hálózatok programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Az értekezés elfogadását javasolom.

Debrecen, 2009……

……..……
Dr. Sztrik János
témavezető

Tartalomjegyzék

1. Bevezetés

A nagytávolságú WAN *(Wide Area Network)* adatkommunikációs hálózatok fizikai sávszélessége a technológiai fejlődésnek köszönhetően egyre dinamikusabban növekszik, átlépve a MAN *(Metropolitan Area Network)* környezetben elterjedten alkalmazott 1 Gbit/sec-os átviteli kapacitást. A fizikai sávszélesség növekedése azonban nem elegendő ahhoz, hogy a nagyságrendbeli átviteli teljesítménytöbbletet alkalmazás-szinten is érzékelhessük. Tudjuk, hogy IP *(Internet Protocol)* hálózaton nagymennyiségű adatot megbízhatóan TCP *(Transmission Control Protocol)* szállítási protokoll segítségével továbbíthatunk. Azonban a TCP torlódásvezérlő *(congestion control)* és ablakméret-szabályzó *(windowing)* mechanizmusa komoly korlátot jelent az effektív átviteli teljesítményre a nagytávolságú és nagy sávszélességű (pl. nemzetközi, interkontinentális) kapcsolatokon. A standard Reno alapú TCP torlódásvezérléséből eredő szűk keresztmetszetre egy gyakorlati példával szeretnénk rávilágítani: Egy 100ms késleltetésű, 10Gbit-es kapcsolaton 1500 bájtos standard TCP csomagokat forgalmazunk. Ahhoz, hogy egy TCP adatfolyammal tartósan elérjük a 10 Gbit-es fizikai kapcsolat maximális hasznos sávszélességét *(net bandwidth)*, az effektív ablakméretet 121665 Kbájt méretűre kellene beállítani, ezen kívül a csomagvesztési arány nem lehetne több mint egyetlen torlódási esemény 5 millió csomagonként[1].

A klasszikus (Reno alapú) TCP protokoll átvitelt vezérlő mechanizmusai a hálózati technológiák egy másik szegmensében, a vezeték nélküli helyi hálózatokban *(WiFi – Wireless fidelity, WLAN – Wireless Local Area Network)* is komoly teljesítménycsökkenést eredményezhetnek. A torlódásvezérlő és az újraküldést szabályzó mechanizmusok természetükből adódóan a mobil WiFi hálózatokban még súlyosabb anomáliákat generálnak. Ebben az esetben mobil hálózat alatt olyan infrastruktúrát értünk, melyben a rádiós cellák közötti fizikai mozgás biztosított a vezetéknélküli kliens számára.

A kialakított tesztrendszerekkel és a benne végzett mérésekkel célom az volt, hogy megvizsgáljam az utóbbi időben megjelent alternatív TCP változatok teljesítményparamétereit, valamint karakterisztikus viselkedésüket nagysebességű, magas késleltetésű (valós) optikai hálózatokon és mobil WiFi környezetben, továbbá kidolgozzak olyan beállítási módszereket, melyekkel hatékonyan befolyásolható az alap algoritmus működése. Korábbi

tanulmányokban találhatunk konkrét alternatív TCP variánson elvégzett összehasonlító elemzéseket, azonban több variánsra kiterjedő komplex, kereszt réteges *(cross-layer)* vizsgálatokat bemutató tanulmányból ezidáig kevés született. Természetesen a korábbi kutatások eredményeiből kiindulva terveztem meg az alkalmazott vizsgálati módszereket. A disszertáció első részében bemutatom azokat az elemzéseket, melyeknél nagysebességű, valamint aszimmetrikus kapcsolatokon vizsgáltam a TCP protokoll működését befolyásoló operációs rendszer kernel-változóit (pl. puffer- és ablakméretek, torlódásvezerlés) először alapértelmezett értékekkel, majd a tapasztalatok alapján módosított paraméter-beállításokkal és módosított algoritmusokkal dolgoztam. A második részben mobil vezeték nélküli környezetben folytattam a kereszt-réteges vizsgálatokat, IPv4 és IPv6 infrastruktúrán, valamint továbblépve megnéztem a TCP viselkedését védett WiFi hálózatban bekövetkezett roaming események kapcsán.

2. Többrétegű forgalom- és protokoll-elemzés

A többrétegű *(cross-layer)* forgalomelemzés egy viszonylag újkeletű elemzési technika, melynek legfőbb célja a kommunikáció során a hálózati rétegekben végbemenő folyamatok egymásra és magára a kommunikációra kifejtett hatásának komplex vizsgáltata. A TCP/IP referenciamodell egyes rétegeiből kinyert információk együttes vizsgálata vezet el a kommunikáció során felmerülő komplex problémák megértéséig. Például vezeték nélküli hálózatok esetén a médium jellegéből adódó relative gyakori bithibák nagyságrendjére a TCP szegmensek vizsgálatából is tudunk következtetni. A többréteges analízis során megfigyelhető, hogyan hatnak az egyes rétegekben bekövetkező események a rétegmodell többi elemére.

3. A TCP szállítási protokoll felépítése

A TCP (*Transmission Control Protocol)* az Internet protokoll-készlet egyik központi protokollja. Végpontok közötti kapcsolatorientált, megbízható és sorrendtartó bájtfolyam szolgáltatást nyújtó szállítási protokoll (a TCP/IP

referenciamodell szállítási rétegében nyújt szolgáltatásokat, köztes interfészként az hálózati (IP) réteg és az alkalmazás között). A kapcsolatorientált működés során az adatátvitel megkezdése előtt a két kommunikáló folyamatnak kapcsolatot kell felépíteniük egymás között. A TCP elvégzi a küldő oldali alkalmazás számára az elküldött bájtfolyam szegmentálását, valamint fogadó oldali összeállítását. A protokoll full duplex, azaz egy TCP kapcsolat egyidőben párhuzamosan mindkét irányba képes bájtfolyam átvitelére. Ezen felül a TCP adatfolyam- *(flow control)* és torlódásvezerlő *(congestion control)* mechanizmusokat is tartalmaz, mellyekkel a fogadó és a küldő oldal képes szabályozni az átvitt adatmennyiséget. A TCP számára fontosabb a megbízható átvitel, mint az időbeni pontosság[2,6].

A két végponton futó folyamatok *(process)* TCP socket-eken keresztűl kommunikálnak egymással. A TCP kapcsolat mindkét oldalán található egy-egy socket, melyeket az IP cím és a hozzás kapcsolt portszám azonosít *< IP_address, port_number>*. A két folyamat között így létrejön egy logikai kapcsolat, melyet a két socket egyedien azonosít: *< local_IP_address, local_port, remote_IP_address, remote_port>*.

3.1. A TCP protokoll szolgáltatásai

3.1.1. Megbízhatóság

A protokoll minden elküldött bájthoz szekvenciaszámot rendel, melyre megérkezés után pozitív nyugtázást *(ACK)* vár a fogadó TCP entitástól. Amennyiben a nyugta nem érkezik meg egy meghatározott időintervallumon belül, úgy a TCP újraküldi az adatot. A fogadó oldali TCP entitás a szegmensek sorbarendezéséhez használja a szekvencia számozást, abban az esetben, ha a sorrendtartás felborulna, vagy duplikáció merülne fel.

3.1.2. Adatfolyam átvitel

Az alkalmazás szemszögéből a kommunikáció egy folytonos bájtfolyamként jelenik meg, melyet a TCP szegmensekre bont és megfelelő protokoll-információval ellátva továbbít az IP, azaz a hálózati réteg számára. A TCP határozza meg a létrehozott szegmensek aktuális méretét, és továbbításuk ütemezését.

3.1.3. Adatfolyam vezérlés

Amikor a fogadó TCP entitás nyugtát küld vissza a küldőnek, egyben azt is jelzi, hogy az adott pillanatban mennyi adatot képes még fogadni az utolsó megérkezett TCP szegmensen felül anélkül, hogy fogadó oldali puffer túlcsordulás következne be. A nyugtázó üzenet tartalmaz egy felső korlátot, úgy, hogy azt a legmagasabb szekvenciaszámot adja meg, melyet még képes probléma nélkül fogadni.

3.1.4. Multiplexelés

A TCP 16 bites portazonosítokat alkalmaz az egyes folyamatokhoz tartozó kapcsolatok beazonosítására, miáltal lehetővé teszi egy adott hálózati végponton több folyamat párhuzamos TCP alapú kommunikációját. A hálózat- és a csomópontazonosító kombinációjából jön létre a socket azonosító. Egy socket-pár egyedileg azonosítja a TCP kapcsolatot.

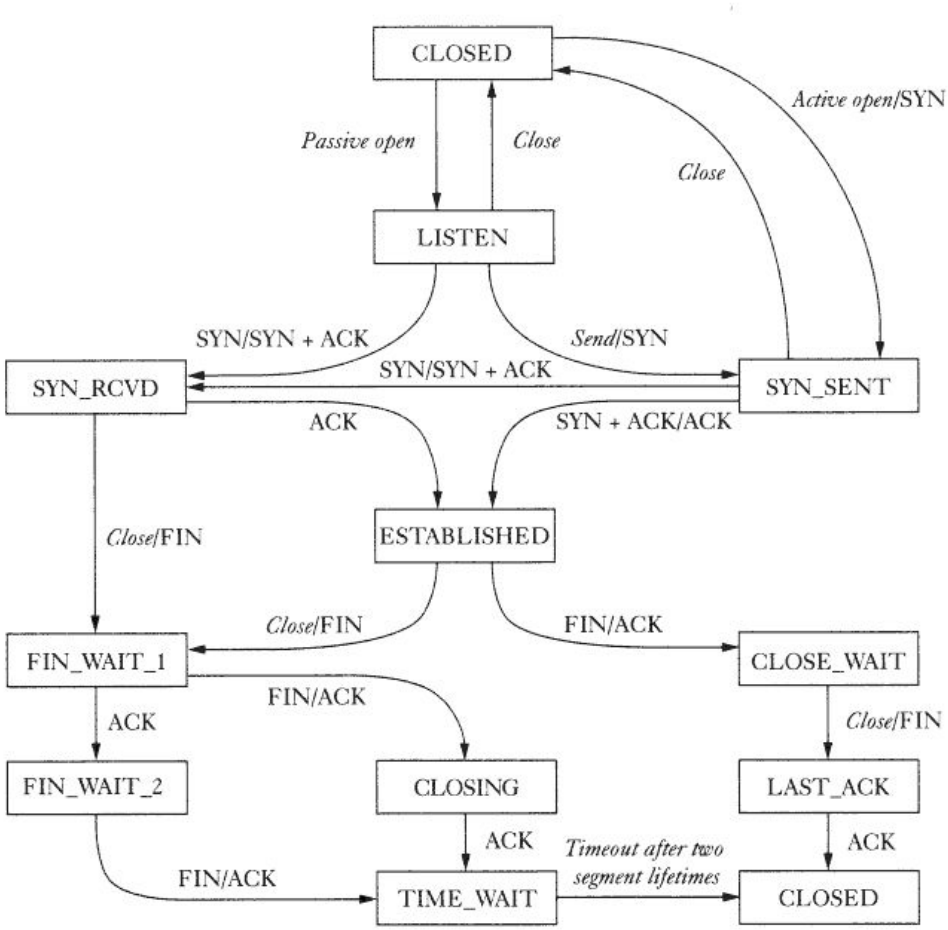
3.1.5. Logikai kapcsolatok

A következőkben ismertetett megbízhatósági és adatfolyam-vezérlő mechanizmusok számára a TCP minden adatfolyamhoz státusz információkat rendel. Logikai kapcsolatnak nevezzük a státuszinformációk, a socket pár, az ablakméretek, és a szekvencia számok együttesét.

3.1.6. Full Duplex átvitel

A TCP kétirányú, full duplex adatfolyam-kezelést biztosít.

A TCP állapotátmenet diagramja



1. ábra A TCP kapcsolat állapotátmenetei

A nyitott kapcsolathoz, azaz az ESTABLISHED állapothoz két átmeneten keresztül juthatunk el, illetve két átmenet vezet az ESTABLISHED állapotból a kapcsolat befejezéséhez. Bármely irányban adatátvitel csak ESTABLISHED állapotban mehet végbe.

3.2. Háromlépéses kézfogás (3-way handshake)

Amennyiben a kapcsolat LISTEN állapotban van, és érkezik egy SYN szegmens, akkor a kapcsolat átmegy SYN_RCVD állapotba és SYN+ACK szegmenssel válaszol a kérelmezőnek. A kliens oldal ebben az esetben aktív kapcsolatnyitást hajt végre, a kiküldött SYN szegmens hatására átkerül a kapcsolat SYN_SENT állapotba. A túlsó féltől érkező SYN+ACK válasz után ESTABLISED állapotba jut és a 3-way handshake algoritmus harmadik lépéseként egy ACK üzenetet küld a másik félnek. Amikor ez a nyugta is megérkezik, akkor került át a kapcsolat távoli fele is ESTABLISHED állapotba[2,6].

A kapcsolat befejezésénél különös figyelmet kell fordítani arra, hogy a végponton a folyamatnak a másik féltől függetlenül be kell zárnia a TCP kapcsolatot a saját oldalán. Következésképpen bármely oldalon három féle állapotátmenettel juthatunk el ESTABLISHED állapotból CLOSED állapotba:

- Elsőként a saját oldal zárja a kapcsolatot:

  ESTABLISHED -> FIN_WAIT_1-> FIN_WAIT_2 -> TIME_WAIT -> CLOSED.

- Elsőként a másik oldal zárja a kapcsolatot:

  ESTABLISHED -> CLOSE_WAIT -> LAST_ACK -> CLOSED.

- Mindkét oldal egyidőben zárja a kapcsolatot:

  ESTABLISHED -> FIN_WAIT_1-> CLOSING ->TIME_WAIT -> CLOSED.

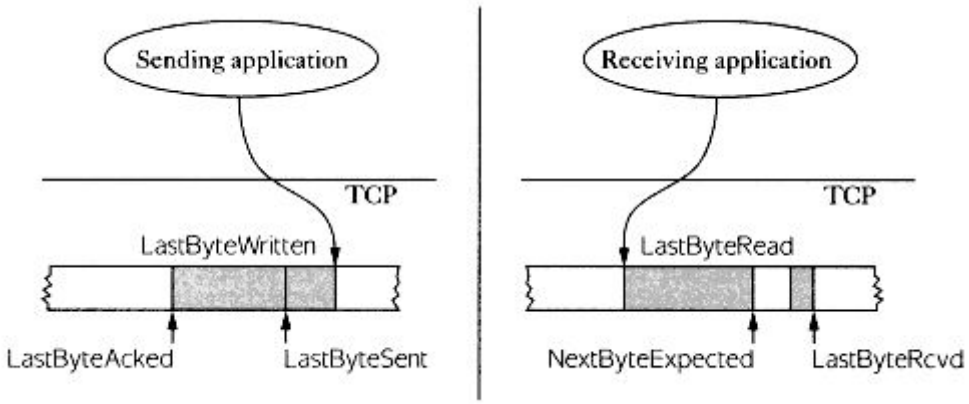
A legfontosabb szempont a kapcsolat bontásakor, hogy a TIME_WAIT állapotban lévő kapcsolat csak akkor kerülhet át CLOSED állapotba, ha az IP csomag TTL (Time-To-Live) értékének kétszeresével megegyező ideig várakozott. Ennek magyarázata, hogy a helyi oldal nyugtát küld a távoli oldal FIN szegmensére válaszul, ugyanakkor azt nem tudhatja, hogy meg is mérkezik a másik félhez. Következésképpen a másik fél újraküldi a FIN szegmenst, melynek átvitelét a hálózat számos okból kifolyólag késleltetheti. Abban az esetben, ha a kapcsolatnak megengednénk, hogy közvetlenül CLOSED állapotra váltson, úgy előállhatna az a szituáció, hogy a bezárás után tetszőleges másik alkalmazásr új kapcsolatot épít ki ugyanazon a porton keresztül, és a korábbi

kapcsolatból fennmaradt, késleltetett FIN szegmens ekkor érkezik be. Ebben az esetben az új kapcsolat azonnal terminálódik.

3.3. Sliding window

A sliding window számos megbízhatósági szolgáltatást nyújt:

1) Garantált, megbízható adatátvitel,
2) Sorrend tartó átvitel,
3) Adatfolyam vezérlés a küldő és a fogadó végpont között.



2. ábra Bájtfolyam kezelése a szállítási rétegben

3.3.1. Megbízható és sorrendtartó átvitel

A TCP küldő és fogadó oldala az alábbiak szerint kommunikál a megbízható és sorrendtartó átvitel érdekében:

- Minden bájthoz hozzárendelődik egy szekvencia szám.
- A nyugták kommulatívak.

Küldő oldal

- Utolsó_nyugtázott_bájt <= Utolsó_elküldött_bájt
- Utolsó_elküldött_bájt <= Utolsó_kiírt_bájt
- Az *Utolsó_nyugtázott_bájt* és az *Utolsó_kiírt_bájt* közötti bájtokat pufferelni kell.

Fogadó oldal

- Utolsó_beolvasott_bájt < Következő_várható_bájt
- Következő_várható_bájt <= Utolsó_fogadott_bájt + 1
- A *Következő_beolvasott_bájt* és az *Utolsó_beérkezett_bájt* közötti bájtokat pufferelni kell.

Flow Control

- Küldő puffer mérete: MaxSendBuffer
- Fogadó puffer mérete: MaxRcvBuffer

Fogadó oldal

- Utolsó beérkezett bájt - Következő olvasott bájt <= MaxRcvBuffer
- Meghirdetett ablak = MaxRcvBuffer - (Utolsó beérkezett bájt - Következő beolvasott bájt)

Küldő oldal

- Utolsó elküldött bájt – Utolsó nyugtázott bájt <= meghírdetett ablak
- Effectív ablak = Meghirdetett ablak - (Utoló elküldött bájt - Utolsó nyugtázott bájt)
- Utolsó kiírt bájt - Utolsó nyugtázott bájt <= MaxSendBuffer
- A küldő blokkolása ha (Utolsó kiírt bájt - Utolsó nyugtázott bájt) + y > MaxSendBuffer

A fogadó oldal minden esetben nyugtázza a megérkezett szegmenseket. Amikor a fogadó oldal kapcsolat-puffere telítődik, nulla bájt méretű ablakot hirdet a küldő számára, aki ennek hatására felfüggeszti a szegmensek küldését.

3.4. Adaptív újraküldés

A TCP megbízható átvitelt garantál, így újraküldi a szegmenst abban az esetben, ha a nyugta nem érkezik vissza egy adott időintervallum alatt. Ezt az időtúllépési értéket a kapcsolat két végpontja közötti effektív RTT függvényében állítja be a TCP. Azonban az Internet két tetszőleges csomópontja közötti lehetséges RTT tartományt, valamint annak időbeni varianciáját ismerve sem könnyű a megfelelő időtúllépési érték meghatározása. A probléma megoldásához adaptív újraküldési algoritmust alkalmaz a protokoll. A következőkben ismertetjük az algoritmus működését és időbeni fejlődését[6].

Eredeti algoritmus

Megméri a körbefordulási időt (sample Round Trip Time) minden egyes szegmens/nyugta párra, és súlyozott átlagot számol rá.

$$Becsült\_RTT = a \times Becsült\_RTT + b \times Vételezett\_RTT,$$

$$ahol\ a + b = 1$$
$$0{,}8 <= a <= 0{,}9$$
$$0{,}1 <= b <= 0{,}2$$

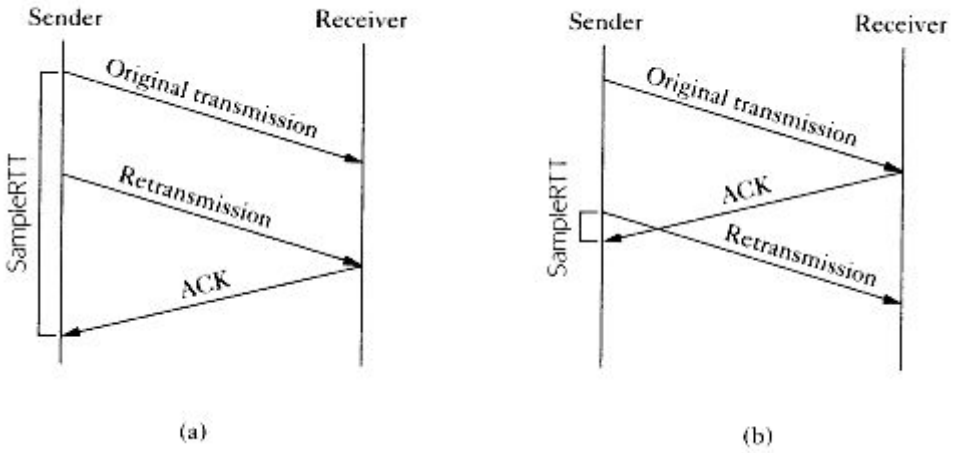
Az időtúllépést a becsült RTT alapján állítja be:

$$Időtúllépés = 2 \times Becsült\_RTT$$

Karn/Partridge algoritmus

- Nem méri az RTT-t újraküldéskor
- Kétszeres időtúllépési határérték beállítása újraküldéskor

Jacobson/Karels algoritmus



3. ábra Az algoritmus sematikus működése

Az átlagos RTT újraszámítása:

Különbség = Vételezett_RTT – Becsült_RTT

Becsült_RTT = Becsült_RTT + (d x Különbség)

Eltérés = Eltérés + d (|Különbség| – Eltérés), ahol d egy 0 és 1 közötti valós szám

Figyelembe veszi a varianciát a túllépési érték beállításánál:

Időtúllépés = u x Becsült_RTT + q x Eltérés, ahol u = 1 és q = 4

3.5. Torlódásvezérlés

Lassú indítás (Slow start)

Működés közben az algoritmus gondoskodik arról, hogy a csomagok elküldési rátája megegyezzen a nyugták visszaérkezési rátájával.

A slow start mechanizmus egy speciális ablakot definiál a küldő oldali TCP entitás számára, melyet torlódási ablaknak *(congestion window – cwnd)* nevezünk. Kapcsolat létesítésekor a torlódási ablakot 1 szegmens méretével megegyező kezdeti értékre állítja be (ami lehet a fogadó fél által hirdetett

szegmensméret, vagy egy alapértelmezett érték, tipikusan 512 bájt). Amikor nyugta érkezik a fogadótól, a torlódási ablak egy szegmenssel növekszik. A maximális effektív átviteli ráta a torlódási és a meghirdetett ablakméretek minimuma lehet. Tehát a torlódási ablak a küldő oldali adatfolyam-szabályozás, míg a meghirdetett ablak a fogadó oldali adatfolyam-szabályozás eszköze. Előbbi a küldő által észlelt hálózati torlódás, utóbbi a fogadó oldalon rendelkezésre álló pufferméret függvénye.

A küldő fél kezdetben egyetlen szegmenst küld el, majd nyugtázásra várakozik. Amint megérkezik a nyugta a torlódási ablakot két szegmens méretére emeli, azaz két szegmenst küldhet ki nyugtázatlanul. Amint nyugtázásra kerül mindkettő, a torlódási ablakot négy szegmens méretére emeli. Így exponenciális növekményt kapunk. Valójában nem pontosan exponenciálist, minthogy a fogadó késleltetheti a nyugtázást, jellemzően oly módon, hogy nyugtát küld minden második megérkezett szegmens után[4].

Adott esetben a TCP kapcsolat terhelése elérheti a hálózat kapacitását, ilyen helyzetben a két végpont közötti IP útvonalon bármely forgalomirányító eldobhat csomagokat. A küldő oldal ebből érzékeli, hogy a beállított torlódási ablakméret túl nagy.

A korai TCP implementációk csak akkor indították az átvitelt slow start algoritmus szerint, ha a fogadó fél nem a saját IP hálózatukban volt. A mai implementációk minden esetben alkalmazzák a slow start mechanizmust[3].

Torlódás-elkerülés

Torlódás következhet be, amikor egy nagysávszélességű (tipikusan LAN) hálózatról halad át a csomag egy alacsonyabb sávszélességű (tipikusan WAN) hálózatra. Akkor is bekövetkezhet torlódás, amikor több bemenő adatfolyam érkezik egyidőben a forgalomirányítóhoz, melynek kimeneti kapacitása kisebb, mint a bemenő adatfolyamok rátáinak összege. A torlódás-elkerülés egy lehetésges módja a csomagvesztés megelőzésének, kezelésének[1,4].

Az algoritmus feltételezi, hogy a bithibából eredő csomagvesztés aránya nagyon alacsony (jóval 1% alatt van), tehát a csomagvesztés azt jelzi, hogy valahol a küldő és a fogadó között a hálózaton torlódás lépett fel. Két indikátora van a csomagvesztésnek: időtúllépés és duplikált nyugták érkezése.

A slow start és a torlódás-megelőzés két egymástól független algoritmus, eltérő funkcionalitással. Amikor torlódás következik be, a TCP-nek le kell lassítania az átvitelt, majd a forgalom konszolidálásához aktivizálnia kell a slow start mechanizmust. A gyakorlatban együtt implementálják a két mechanizmust.

A slow start és a tolódás-megelőző mechanizmusok működéséhez két változót kell karbantartani minden egyes TCP kapcsolat számára: torlódási ablakméret (cwnd) és a slow start határérték mérete (ssthresh). A kombinált algoritmus az alábbiak szerint működik:

1. Egy kapcsolat kezdeti beállításakor a torlódási ablak mérete 1 szegmens méretével egyezik meg, míg a slow start threshold 65535 bájt méretű.

2. A TCP kimeneti rutinja küldéskor a torlódási és a fogadó által hírdetett ablakméretek minimumát veszi figyelembe min(cwnd, advwnd) tehát ennél nagyobb mennyiségú adatot nem küld ki nyugtázatlanul.

3. Torlódás esetén (melyet időtúllépés vagy duplikált nyugta érkezése jelez) az érvényes ablakméret ½-e kerül tárolásra (a torlódási és a fogadó által hirdetett ablakméretek minimuma, de legalább két szegmens méretével megegyező érték) az ssthresh (slow start threshold) változóban. Továbbá, ha időtúllépés jelzi a torlódást, a cwnd változót 1 szegmens méretére állítja be a mechanizmus (slow start).

4. Amikor a másik fél új adatot nyugtáz, a küldő növeli a torlódási ablakot. A növelés mértéke függ a TCP aktuális üzemmódjától: slow start vagy torlódás-megelőzés.

Ha a torlódási ablak kisebb vagy egyenlő mint az ssthresh, akkor a TCP slow start üzemmódban van, más esetben a torlódás megelőzési mechanizmusa aktív. A slow start fázis addig tart, míg a cwnd el nem éri az ssthresh értéket, azaz az előző torlódáskor érvényes cwnd ablakméret felét, ezt túllépve a torlódás-megelőzési mechanizmus veszi át a vezérlést[1].

A slow start állapotában a cwnd egy szegmenssel kezd, és minden alkalommal, amikor ACK érkezik, egy újabb szegmenssel növekszik. Ahogy korábban említettem, ez exponenciálisan nyitja meg az ablakot: először egy szegmenst küld, majd kettőt, négyet, és így tovább. A torlódás-megelőzés szabja meg, hogy a cwnd *segsize*segsize/cwnd* értékkel növekedjen minden alkalommal, amikor ACK érkezik, ahol a *segsize* a szegmens mérete, és a cwnd értéke bájtban. A

slow start exponenciális növekedéséhez viszonyitva a cwnd növekedése lineáris. A cwnd növkedésének értéke legfeljebb egy szegmens lehet minden egyes körbefordulási időnként (függetlenül az abban a RTT-ban beérkezett nyugták számától), míg a slow start a körbefordulási idő alatt beérkezett ACK-k számával növeli a cwnd-t, azaz a torlódási ablakot.

Gyors újraküldés

A TCP valós idejű nyugtát (duplikált ACK-t) generálhat, amikor sorrendiséget felborító szegmens érkezik. Ez a duplikált nyugta nem késleltethető. A duplikált nyugta célja, hogy informálja a másik végpontot, hogy a szegmens nem megfelelően érkezett, és hogy jelezze, mennyi a várható szekvenciaszám.

Mivel a TCP nem képes megállapítani, hogy a duplikált nyugta szegmens-vesztés vagy a szegmensek átrendeződése miatt keletkezett, várakozik, amíg egy kisebb számú ACK be nem érkezik. Ha feltehetően csak a szegmensek átrendeződéséről van szó, csak egy vagy két duplikált nyugta érkezik mielőtt az átsorolt szegmens feldolgozásra kerül, ami még egy új nyugtát fog generálni. Ha sorozatban három vagy több nyugta érkezik, egyértelmű jelzés, hogy a szegmens elveszett. A TCP ezután elvégzi a hiányzónak tűnő szegmens újraküldését, nem várja meg az újraküldési időtartam végét.

Gyors helyreállítás

A gyors helyreállítás után a hiányzó szegmenseket elküldi a küldő oldali TCP entitás, és működésbe lép a torlódás-megelőzés, ugyanakkor a slow start nem került végrehajtásra. Ez a gyors helyreállítás algoritmus. A javulás nagy ablakméret esetén látványos, minthogy nagy átviteli teljesítményt biztosít közepesen torlódott környezetben.

Annak magyarázata, hogy ebben az esetben a küldő nem hajt verge slow start-ot, az, hogy a duplikált nyugta beérkezése a csomagvesztés tényénél több információt jelent a TCP számára. Mivel a fogadó csak abban az esetben generálhat másolati nyugtát, amikor újabb szegmens érkezik be hozzá, tehát a szegmens elhagyta a hálózatot, és már a fogadó kapcsolat-pufferében van. Ez jelzi a TCP számára, hogy még van adatátvitel a két végpont között, így nem kell hirtelen slow start üzemmódra váltva csökkentenie a küldési rátát.

A gyors újraküldés és a gyors helyreállítás algoritmusokat is gyakran közös kódban, együtt implementálják, az alábbiak szerint:

1. Amikor egymás után a harmadik duplikált nyugta is megérkezik, az ssthresh értékét beállítja a cwnd érték felére, ami nem lehet kevesebb mint két szegmens. Újraküldi a hiányzó szegmenst, megnöveli a cwnd értéket az ssthresh plusz 3x a szegmensméret értékre. Tehát a cwnd a hálózatra kiküldött szegmensek és a fogadó által letárolt szegmensek együttes méretével megegyező mértékben kerül kiterjesztésre.

2. Amikor másik duplikált nyugta érkezik a cwnd érték a szegmens méretével megegyező mértékben növekszik. Ez a mechanizmus megnöveli a torlódási ablakméretet annyival, hogy az éppen elküldött szegmens is beleférjen. Küldhet újabb szegmenst, ha az újonnan beállított cwnd érték ezt lehetővé teszi.

3. Amikor megérkezik a következő ACK, mely az új adatot nyugtázza, a cwnd az ssthresh értékét veszi fel. Ez az ACK az első lépésben megadott újraküldés nyugtája kell, hogy legyen, egy RTT-vel az újraküldés után. Ezenkívül ennek az ACK-nak minden, az elveszett csomag és az első duplikált ACK megérkezése közötti szegmenst is nyugtáznia kell. Ez a TCP torlódás-megelőző mechanizmusa, minthogy a TCP csomagvesztés esetén a ½-ére állítja be a küldési rátát.

3.6. Alternatív torlódásvezérlő mechanizmusok

TCP Reno

A TCP Reno csomagvesztést indukálva becsli a rendelkezésre álló sávszélességet. Amíg nincs csomagvesztés, addig eggyel növeli az ablakméretet, míg veszteség bekövezésekor a felére csökkenti a torlódási ablakméretet. Ezt az eljárást nevezzük Adaptive Increase Multiplicative Decrease algoritmusnak. Belátható, hogy a Reno-ba beépített torlódás elkerülési mechanizmus az ablak méret periodikus ingadozását eredményezi a folyamatos ablakméret változtatás hatására[8]. Az ablakméret ingadozása a csomagok késleltetésének ingadozását vonja maga után. Ez az ingadozás nagyobb késleltetési jittert eredményez,

valamint a rendelkezésre álló sávszélesség nem használható ki hatékonyan a csomagvesztések hatására bekövetkező újraküldések következtében.

cwnd = cwnd + 1

Az ablakméret aktualizálásának üteme függ a kapcsolat késleltetésétől. Az alacsonyabb késleltetésű kapcsolatok ablakmérete gyorsabban növekszik a nagyobb késleltetésű kapcsolatokkal szemben, melynek nemkívánt hatása, hogy a rendelkezésre álló sávszélességből nagyobb arányban részesülnek.

A klasszikus Reno torlódásvezérlés korlátai: alacsony hatékonyság magas BDP értékű hálózatokon, 64K maximális ablakméret, nincs szelektív nyugtázás (SACK)[5].

TCP BIC

A Binary Increase Congestion control a TCP torlódásvezérlő mechanizmusának nagy sebességű, magas késleltetésű hálózatokra optimalizált változata. Bináris kereső algoritmus alkalmaz, mellyel logaritmikusan növelhető az ablakméret[10]. A Linux kernel alapértelmezett TCP torlódásvezérlő mechanizmusa a 2.6.8-tól a 2.6.18-as verzióig.

TCP Cubic

A Cubic a BIC egy kevésbe agresszív, szisztematikusabb változata, melyben az ablak a legutóbbi torlódási esemény óta eltelt idő harmadrendű függvénye, egy inflekszió pontot határoz meg az ablak számára, a torlódás bekövetkezését megelőzően[10]. A Linux kernel alapértelmezett TCP torlódásvezérlő mechanizmusa a 2.6.19 verzióval kezdődően.

HSTCP

A HighSpeed TCP (HSTCP) szintén egy alternatív torlódásvezérlő mechanizmus, melyet az IETF RFC 3649 definiál[11]. Amikor nyugta érkezik az ablak $a(w) / w$ szerint növekszik, majd amikor csomagvesztést detektál az algoritmus három duplikált nyugta megérkezésével az ablak $(1 - b(w))w$ szerint fog csökkenni, ahol w az effektív ablakméret.

Alacsony ablak esetény a Reno-val megegyező módon működik a HSTCP, tehát $a(w) = 1$ és $b(w) = 0.5$. Amikor a torlódási ablak egy bizonyos határérték felett van az $a(w)$ és $b(w)$ az effektív ablakméret függvénye lesz. Ebben a

tartományban amikor növekszik a torlódási ablak, úgy az *a(w)* értéke is növekszik és a *b(w)* értéke csökken. Ez azt jelenti, hogy az ablak gyorsabban nő, mint a TCP Reno esetén, és csomagvesztés esetén is gyorsabban áll vissza. Így a HSTCP barátságosan viselkedik a párhuzamos standard TCP folyamokkal szemben átlagos hálózati környezetben, emellett gyorsan képes felhasználni a rendelkezésre álló sávszélességet magas B x D szorzatú hálózatokon.

TCP Westwood

A protokoll a TCP Reno küldő oldali módosításával kívánja javítani az átviteli teljesítményt olyan nagysebességű útvonalakon, ahol az átviteli hibákból, vagy nagy terhelésből adódó csomagvesztés nagy valószínűséggel bekövetkezhet. A TCPW a nyugtafolyamból nyeri ki a torlódási paraméterek hangoláshoz szükséges információkat: Slow Start Threshold (ssthresh), and Congestion Window (cwnd).

Alkalmazásával különösen hatékonyan növelhető az átviteli teljesítmény nagy hibaarányú vezeték nélküli átviteli közegen. A végpontok közötti útvonal sávszélességének becslésével dolgozik a TCPW, hogy megkülönböztesse a klasszikus csomagvesztési eseményt a rádiós átvitelben bit szinten bekövetkező átviteli hibáktól. Ez a megkülönböztetési képesség teszi hatékonnyá az algotimust. Az alapelv az, hogy küldő oldalon folyamatosan figyeli a visszaérkező nyugtafolyam segítségével a kapcsolat sebességét. A becsült eredményt felhasználja a torlódási ablak és a slow start threshold kiszámításánál torlódási esemény bekövetkeztekor, vagyis három duplikált nyugta megérkezésekor, vagy időtúllépés esetén. A stratégia mögötti logika egyszerű: szemben a TCP Reno-val, mely kvázi vakon, minden információ felhasználása nélkül felezi le az ablakméretet három duplikált nyugta után, TCPW megpróbál olyan cwnd es ssthresh értékeket választani, melyek összhangban állnak a torlódás észlelésekor effektív sávszélességgel. Ezta mechanizmust gyorsabb helyreállításnak nevezzük. A vezeték nélküli kapcsolatokon szórványosan keletkeznek bitszintű átviteli hibák, melyeket a standard TCP változat gyakran csomagvesztési eseményként értelmez, ami szükségtelen ablakcsökkentést és ezáltal teljesítménycsökkenést eredményez[9].

Scalable TCP

A torlódási ablakot szabályzó algoritmus küldő oldali módosításával javítja az átvitel performanciáját nagy B x D szorzatú hálózatokon. A mechanizmus működése alacsony ablakméretnél a hagyományos TCP stack-kel azonos[12].

A hagyományos TCP probing ideje a küldési rátával, valamint az RTT-vel arányos, a Scalable TCP probing ideje viszont csak az RTT-vel áll arányban, miáltal skálázhatóvá teszi a sémát nagy sebességű hálózatokon.

cwnd = cwnd + 0.01	minden megérkezett nyugta után, amíg nincs csomagvesztés
cwnd = 0.875 * cwnd	csomagvesztés esetén

TCP Hybla

A TCP Hybla célja, hogy a nagy késleltetésű földközeli, illetve műholdas rádiós linkeken forgalmazó TCP kapcsolatok teljesítményét hátrányosan befolyásoló magas RTT hatását kiküszöbölje. A torlódási ablak dinamikájának analitikus vizsgálata alapján módosítja a TCP paramétereket, hogy megszüntesse az átviteli teljesítmény RTT függőségét.

4. TCP vizsgálata aszimmetrikus kapcsolatokon

A kialakított helyi kiterjedésű teszthálózatokban, valamint valós WAN hálózati környezetben végzett mérésekkel célunk az volt, hogy megvizsgáljunk a nagyméretű TCP folyamok karakterisztikáját befolyásoló rétegprotokollokat és azok alkamazási szinten érzékelhető, a szolgáltatás teljesítményére gyakorolt hatását nagysebességű, magas késleltetésű hálózatokon. Vizsgálataink során a TCP/IP referenciamodell rétegeit, illetve a hozzájuk kapcsolódó rétegprotokollokat első lépésben önállóan, majd a teljes adatfolyamot egységében, a rétegek közötti összefüggéseket figyelembe véve elemeztük[J4].

A fejezet második részében megoldásokat adunk a vizsgálat során kimutatott – az algoritmusok működéséből adódó – szűk keresztmetszetek eliminálására.

4.1. Alkalmazott módszerek

A probléma gyökerét az ablak-alapú *(window-based)* torlódásvezérlés átviteli teljesítmény szabályozásában találjuk. Tudjuk, hogy a TCP torlódási ablak méretét *(congestion window)* az elküldött csomagokra adott nyugták érkezési ideje szabályozza, amely elsősorban a hálózat késleltetésének *(RTT – Round Trip Time, latency)* és aktuális terheltségének függvénye. Tehát minél magasabb értékű a sávszélesség és a késleltetés szorzata egy adott kapcsolaton, az algoritmus annál kevésbé képes hatékonyan kihasználni a rendelkezésre álló sávszélességet. Az alkalmazott klasszikus AIMD *(Additive Increase, Multiplicative Decrease)* mechanizmus viszonylag lassan növeli a torlódási ablakméretet, ezáltal az átviteli ráta lassan közelíti a rendelkezésre álló sávszélességet. Ezek mellett fontos megemlíteni, hogy a TCP önszabályzó *(self-clocking)* mechanizmusának működése függ az adott fizikai kapcsolaton egyidejűleg jelenlévő TCP adatfolyamok számától is.

Ezek tudatában megállapítható, hogy ahhoz, hogy egy TCP variáns széles körben, heterogén hálózati környezetben alkalmazható legyen, egyszerre kell megfelelnie mind a nagy átviteli teljesítmény *(performance)*, mind az adatfolyamok közötti méltányosság *(fairness)* kritériumainak. A torlódásvezérlő mechanizmusokat e két szempont egyidejű figyelembevételével, valós hálózati környezetben elemeztük. A vizsgálatokba bevont TCP változatok rövid ismertetése a 3.6 alfejezetben olvasható.

Az ADSL *(Asymmetric Digital Subscriber Line)* egy adatkommunikációs technológia, mely rézérpáras telefonvonalon nyújt a hagyományos analóg (hang-sávú) modemnél nagyságrenddel gyorsabb adatátvitelt. A technológia asszimmertrikus jellegéből adódóan a rendelkezésre álló sávszélesség az egyik irányba nagyobb *(downlink)*, mint a másikba *(uplink)*. Technikai okokkal indokolható az asszimmertrikus sávszélesség nagy távolságú kapcsolatokon. A kapcsolat szolgáltatói végén nagyobb valószínűséggel jelentkezik áthallás a befutó vonalak *(local loops)* között, mivel ezek nagysűrűségű interfészeken aggregálódnak egy adott DSLAM *(Digital Subscriber Line Access Multiplexer)* eszközön. Ezáltal a felfeléirányú jel erőssége a helyi hurok legzajosabb pontján lesz a leggyengébb. További korlátozó tényező a felfeléirányú folyam számára definiált frekvenciatartomány, mely jóval keskenyebb, mint a lefeléirányú folyamé. Az említett paraméterek nyilvánvalóan alacsonyabb felfeléirányú sávszélességet tesznek lehetővé. Alapvetően kép paraméter határozza meg a TCP teljesítményét; a fogadó oldali buffer mérete és a küldő oldali torlódásvezérlés dinamikája, mely a vonali késleltetés függvénye.

4.2. Új eredmények

Alap vizsgáltatokkal kimutatjuk, hogyan korlátozza bizonyos adatforgalmi helyzetekben a felfeléirányú sávszélesség a TCP átviteli teljesítményét. Mint ismeretes a TCP torlódásvezérlő mechanizmusának elsődleges célja a rendelkezésre álló sávszélesség hatékony kihasználása, a csomagvesztés minimalizálása. A TCP torlódásvezérlése valójában egy önszabályzó mechanizmus, ahol a visszaérkező nyugták *(ACK – acknowledgement)* alapján történik a soronkövetkező szegmens elküldése. Amennyiben egy adott ideig puffer FIFO-kban várakozik a nyugta az útvonal bármely pontján, úgy késleltetett beérkezés következik be. Ekkor az egymást követő nyugták beérkezési időpontjai közelebb kerülhetnek egymáshoz (ACK burst), ami a küldő oldalán félrevezeti a TCP önszabályzó mechanizmusát, miáltal az több adatot küldi ki, mint amennyit a hálózat továbbítani képes. A nyugták burst-ös megérkezése torlódási jelenséghez és jelentős teljesítménycsökkenéshez vezet.

Tesztkörnyezet

- Hálózati kommunikációs végpont, mely 1024/128 Kbit/sec sávszélességű ADSL kapcsolattal rendelkezik a szolgáltató DSLAM eszközéig.
- Szervergép 100 Mbit/sec full-duplex helyi hálózati kapcsolattal. 1 Gbit/sec közvetett gerinckapcsolati sávszélesség a szolgáltató *(Internet Service Provider)* és az egyetemi hálózat között.
- A merési adatok karakterisztikája:
  - Nagyméretű bináris állományok átvitele FTP *(File Transfer Protocol)* protokollon keresztül, mindkét irányba
- Operációs rendszer: Fedora Linux Core 6, kernel v2.6.22-5
- FTP szerver: Pure-FTPd v1.0.21-12
- Forgalomelemző alkalmazás: Ethereal v0.99.0, tcpdump

A Linux kernel a 2.6.13 verziótól kezdve támogatja a dinamikusan beépülő *(pluggable)* TCP torlódásvezérlő alkalmazását, ennek megfelelően az elvégzett vizsgálatok ezen a technikai lehetőségen alapulnak. Alaphelyzetben a kernel boot-időben állítja be a TCP memóriapufferek paramétereit. A mérések során viszont a hálózati kapcsolat fizikai sávszélességét és késleltetését vettük figyelembe számításainkhoz. Az optimális memóriapuffer-méretek beállításához meghatároztuk az alkalmazott fizikai kapcsolat B x D (bandwidth x delay) szorzatát a fenti tulajdonságok ismeretében. Az alábbi kernel változókat átállítottuk számításainknak megfelelően:
tcp_mem, tcp_rmem, tcp_wmem, tcp_app_win, tcp_sack, tcp_wmax, tcp_rmax.

A helyi hurokra és a teljes útvonalra kiszámolt BDP értékek:

Megközelítőleg 20 msec vonali késleltetést mértünk az ADSL kapcsolaton *(D, RTT)* a modemtől a DSLAM eszközig, míg ~30 msec-et a teljes útvonalon. A maximális fizikai sávszélesség *(B)* 1024 Kbit/sec volt lefelé irányban és 128 Kbit/sec felfelé irányban.

B x DL = 128 KB x 0.02 sec = 2,56 Kbájt (helyi DSL kapcsolat)

B x DP = 128 KB x 0.03 sec = 3,84 Kbájt (teljes útvonal a kliens és a szerver között)

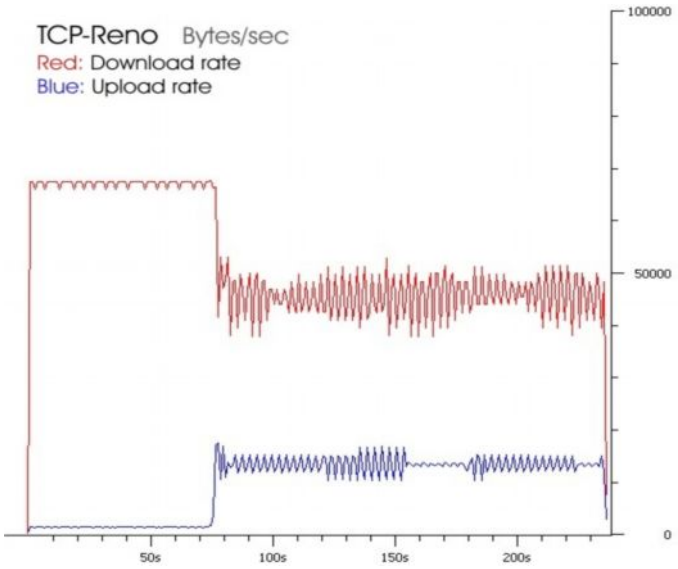
Az alábbi általános szabály megmutatja a BDP szozatból meghatározható optimális TCP fogadó oldali puffer méretét:

Fogadó puffer >= RTT x BW

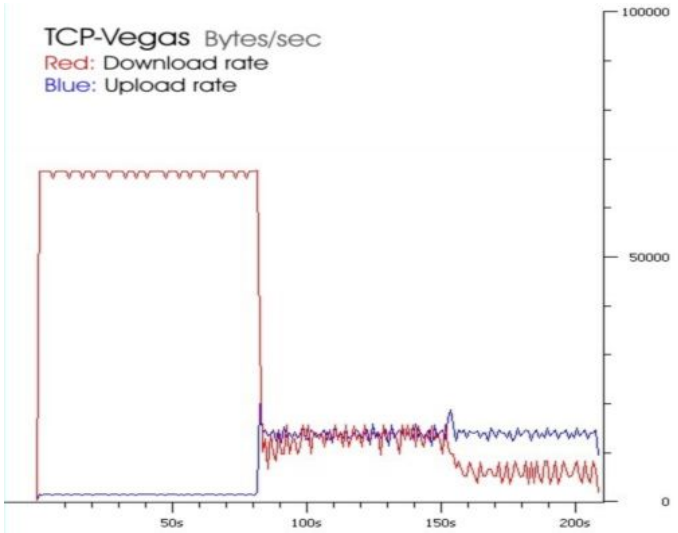
Az alapértelmezett maximális pufferméretek (tipikusan 64 Kbájt) messze magasabbak az általunk kalkulált BDP értékeknél, ezért minden TCP paramétert meghagytunk az alapértelmezett értékén, továbbá aktivizáltuk a szelektív nyugtázás *(selective ACK)* funkciót.

Tesztünk négy fázisból épült fel. Mindegyik – 30 másodperc hosszúságú – fázisban TCP alapú FTP forgalmat generáltunk a szerver és a kliens között. Egyetlen TCP folyammal elértünk a lefeléirányú kapcsolat 100% kapacitását (1. fázis). Ekkor felfelé irányba kizárólag a TCP folyamhoz tartozó nyugták kerültek továbbításra. A második fázisban elindítottunk egy független TCP folyamot az ellenkező irányba, tehát a klienstől a szerver felé. Ennek következtében a felfelé irány gyorsan elérte a 100%-os terheltséget. A 3. és 4. fázisban további két független TCP folyamot indítottunk a kliens felől, ezért, hogy a felfelé irányt minél jobban túlterheljük.

A TCP teljesítményét egyértelműen meghatározza a nyugták beérkezési ideje. Az erősen torlódott felfeléirányú kapcsolaton a nyugták viszont csak késleltetéssel továbbíthatóak. Ebben a fázisban a lefelé irányű TCP folyam átviteli sebessége a rendelkezésre álló fizikai sávszélesség 70%-a körüli értékre csökken, az alkalmazott torlódásvezérlés függvényében. A kliens irányából újabb TCP folyamot indítottunk, még erőteljesebb torlódási jelenséget generálva a felfeléirányú kapcsolaton, ami további átviteli-teljesítmény csökkenést eredményezett a lefeléirányú TCP folyamon (3. fázis). Az effektív átviteli ráta ekkor 40% körüli értéket vett fel. Ebben a helyzetben az erősen torlódott, eredendően is alacsony sávszélességű felfelé irány következtében a rendelkezésre álló lefeléirányú sávszélességet a TCP nem képes hatékonyan kihasználni. Így kimutattuk, hogy a felfelé irány alacsony sávszélessége közvetlen hatással van a lefeléirányú TCP folyam átviteli teljesítményére[J4].



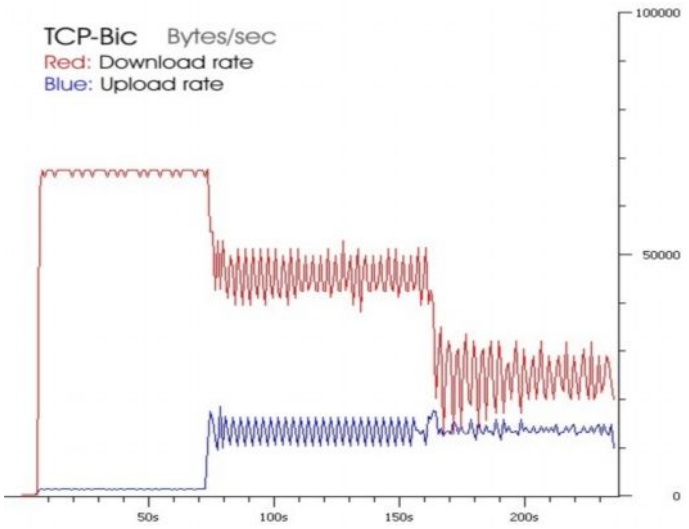
4. ábra TCP Reno teljesítmény



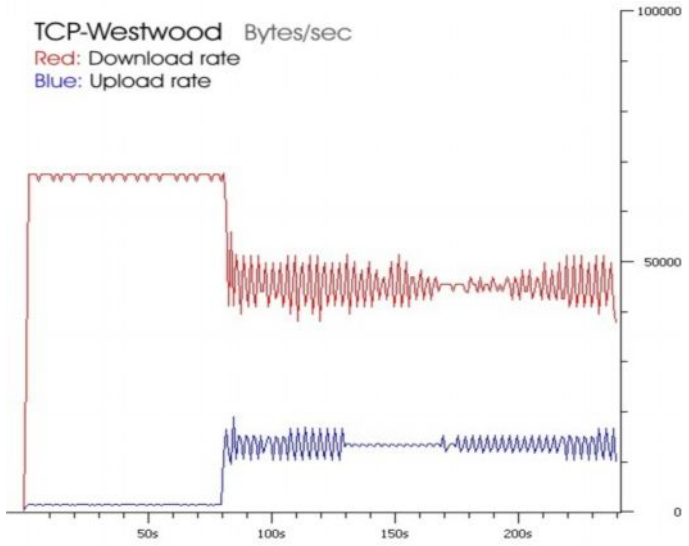
5. ábra TCP Vegas teljesítmény

A vizsgált torlódásvezérlő mechanizmusok a várakozásnak megfelelően közel hasonlóan viselkedtek az első fázis alatt, míg a további fázisok már kimutattak különbségeket. Figyelembe véve az ADSL kapcsolatok

asszimmetrikus jellegét, azok az aktív végfelhasználók, akik jelentős mennyiségű felfeléirányú forgalmat generálnak (video konferencia, Voice over IP, nagyméretű Email, stb.), szembekerülhetnek a TCP teljesítményproblémával, olyan esetekben is, amikor a lefelé irány nem mutat torlódást.

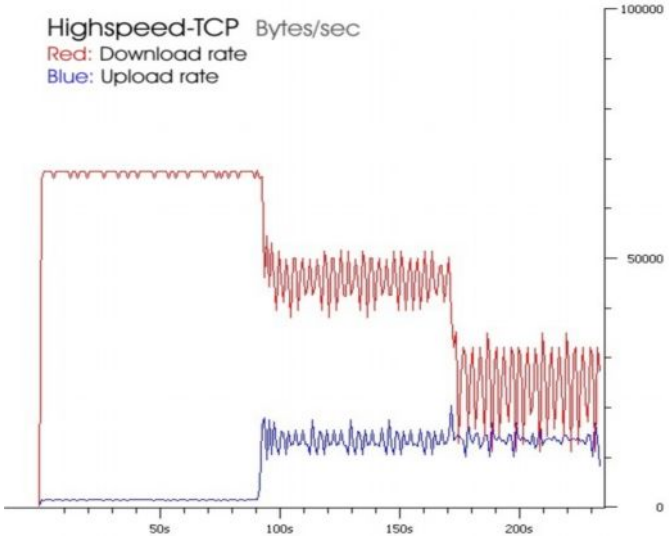


6. ábra TCP BIC teljesítmény



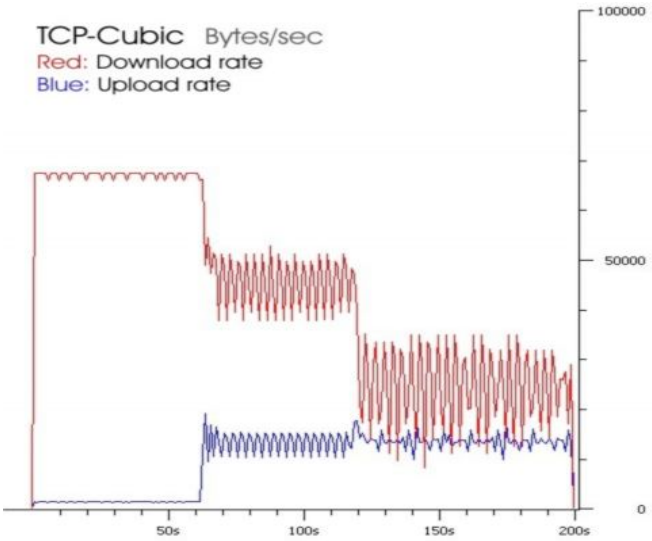
7. ábra TCP Westwood teljesítmény

Egy szélsőséges példát említsunk: a TCP Vegas downstream átviteli teljesítménye messze a legalacsonyabb értéket mutatja, amint megjelenik egy TCP folyam a felfelé irányú kapcsolaton. Ebben az esetben a downlink teljesítmény az uplink átviteli teljesítménye alá is eshet. Tudjuk, hogy a Vegas folyamatos RTT vizsgálattal hangolja az effektív ablakméretet. Torlódást mutató uplink esetén a nyugta csomagok érkezési ideje elég széles intervallumban ingadozik, ami félrevezeti a Vegas algoritmusát. A klasszikus Reno algorimus és annak változatai csak viszonylag lassan képesek reagálni az aktuálisan rendelkezésre álló sávszélességre, míg a korszerű nagysebességű TCP variánsok jóval aggresszívebb ablaméret növelő algoritmussal rendelkeznek *(1., 3., 4. ábra)*.



8. ábra Highspeed TCP teljesítmény

Az agresszív ablakméret növelés következménye: amint torlódás alakul ki a felfelé irányon, úgy mutat az átvitel egyre nagyobb fluktuációt (3. fázis) a nagysebességű TCP variánsoknál.



9. ábra TCP Cubic teljesítmény

A teljesítménycsökkenésen túl az eredendően lapos átviteli karakterisztikával rendelkező downstream TCP folyam irregulárissá válik, ami a hálózaton kommunikáló időérzékeny alkalmazások esetén (streaming, hang és video beszélgetések) megbízhatatlan szolgáltatási minőséget eredményez.

4.3. Következtetés az eredményekből

1. Tézis: Az ADSL kapcsolatok nagyon alacsony sávszélességű uplink-je általános forgalmi jellemzők mellett is könnyen túlterhelhető, mely a lefelé irányú TCP folyam átviteli teljesítményét negatívan befolyásolja. Az uplink torlódás hatására a TCP önszabályzó mechanizmusa lecsökkenti az effektív ablakméretet. További problémát okoz a nyugták érkezési időközeinek csökkenése *(ACK compression)* az uplink-en, ami torlódást okoz a lefelé irányon. Ebben a helyzetben a TCP nem képes hatékonyan kihasználni a fizikai downlink sávszélességet. Következésképpen kimutattuk, hogy a felfelé irányú kapcsolati terheltség közvetlen hatással van a lefelé irányú TCP folyam teljesítményére, továbbá az időérzékeny alkalmazások szolgáltatás-minőségére.

A TCP Vegas ablakméret növelő függvényének nincs pozitív hatása asszimmetrikus kapcsolatokon, mint láttuk, torlódásos uplink esetén

kritikusan alacsony hatásfokú downstream TCP teljesítményt eredményezett. Azt láttuk, hogy a HSTCP, a BIC és Cubic algoritmusok agresszív ablakméretnövelő függvényüknek köszönhetően jelentős átviteliteljesítmény fluktuációt mutattak, amint az uplink-en torlódás jelentkezett. Bár az átviteli teljesítmény átlagát tekintve nagyságrenddel jobb eredmény értek el elődeiknél, az idő-kritikus kommunikáció alkalmazási rétegben tapasztalható szolgáltatás minősége a korszerű variánsok alkalmazásakor is romlott (pl. IP telefónia).

A hálózati kapcsolat sávszélességének és késleltetésének megfelelően számoljuk ki a kapcsolat B x D szorzatát ahhoz, hogy optimáls TCP pufferméreteket alakítsunk ki. Az alacsony uplink sávszélesség forglami szituációk egy jól meghatározott körénél továbbra is jelentős korlátként van jelent az asszimmetrikus kapcsolatokon[J4]. Az aszimmetria az otthoni felhasználóknak szánt internet elérések többségénél technológiától függetlenül még jóideig meghatározó tényező marad, amellett, hogy a jövőben a lefelé és a felfelé irányok közötti sávszélesség különbségek mértéke csökkenést fog mutatni.

5. TCP vizsgáltata nagysebességű, magas késleltetésű hálózatokon

Az aszimmetrikus DSL vonalakon végzett vizsgáltatok tapasztalatai alapján megalkottunk egy tesztkörnyezetet, melyben nagysebességű, magas késleltetésű WAN kapcsolatok TCP torlódásvezérlésre gyakorolt hatását elemeztük. Korábbi cikkekből már tudjuk, hogy egy adatkommunikációs kapcsolat magas késleltetése és nagy sávszélessége együttesen a TCP effektív átviteli teljesítményére nézve komoly korlátot jelent. Pontosabban fogalmazva, a mai nagytávolságú – tehát magas késleltetésű – hálózati technológiák sávszélességének növekedésével a TCP szabályzó mechanizmusai nem képesek lépést tartani. Kérdésként merülhet fel bennünk, hogy miért szeretnénk nagytávolságú WAN kapcsolatokon az adott vonal fizikai sávszélességét megközelítő nagy átviteli teljesítmény elérni egyetlen TCP folyammal, holott az említett kapcsolatok tipikusan aggregált nemzetközi forglamat bonyolítanak le, így forgalom-karakterisztikájuk nagyszámú, egyidejű TCP folyamból tevődik össze.

Itt fontos megemlíteni, hogy a TCP self-clocking mechanizmusának szabályozó hatása az adott fizikai kapcsolaton egyidejűleg jelenlévő TCP adatfolyamok számától is nagymértékben függ. Az egyidejű TCP folyamok folyamatosan versenyeznek a rendelkezésre álló sávszélességért.

Ahhoz, hogy egy TCP variáns széles körben alkalmazható legyen, egyszerre kell megfelelnie mind a nagy átviteli teljesítmény (gyorsan képes alkalmazkodni az aktuálisan rendelkezésre álló sávszélességhez), mind az adatfolyamok közötti méltányosság *(fairness)* kritériumainak. A torlódásvezérlő mechanizmusokat e két szempont egyidejű figyelembevételével, valós hálózati környezetben elemeztük. A vizsgálatokba bevont TCP változatok rövid ismertetése a 3.6 alfejezetben olvasható.

5.1. Új eredmények

A kapcsolat B x D szorzata alapján ebben a tesztsorozatban célunk az volt, hogy az alapértelmezett TCP kernel-változók hangolásával javítsuk a TCP kapcsolat átviteli karakterisztikáját. A teszteket a ma ismert jelentősebb TCP variánsokon végetük el, első alkalommal alapértelmezett beállításokkal, majd a finomhangolt értékek felhasználásával[C7].

A teljes IP útvonalra kiszámolt BDP érték:

Megközelítőleg 4,5 msec átlagos vonali késleltetést mértünk a Debrecen-Budapest optikai kapcsolaton *(D, RTT)* az egyetemi gerinceszköztől a budapesti központi forgalomirányítóig, míg ~4.9 msec-et a teljes útvonalon. A maximális fizikai sávszélesség *(B)* 1 Gbit/sec volt lefelé irányban és 1 Gbit/sec felfelé irányban.

B x D = 125Mbájt x 0.0045 sec = 5625 Kbájt

Az alábbi általános szabály mutatja a BDP szozatból meghatározható optimális TCP fogadó oldali puffer méretét:

Fogadó puffer >= RTT x BW

Az alapértelmezett maximális pufferméretek (64 Kbájt) messze alacsonyabbak az általunk kalkulált BDP értéknél, ezért minden TCP paramétert finomhangoltunk a számított értékeknek megfelelően, továbbá aktivizáltuk a szelektív nyugtázás *(selective ACK) és* az időbélyegzés *(timestamping)* funkciókat. Az alábbi kernel változókat áthangoltuk számításunknak megfelelően:

- *tcp_mem,*
- *tcp_rmem,*
- *tcp_wmem,*
- *tcp_app_win,*
- *tcp_sack,*
- *tcp_wmax,*
- *tcp_rmax*.

Az eredményeket az alábbi két táblázat foglalja össze:

	BIC	**Cubic**	**Reno**	**Vegas**	**HTCP**	**Westwood**	**Scalable**	**Hybla**
TCP paraméterek	Maximális átviteli teljesítmény (Mbit/sec, net bandwidth)							
Default	306	302	284	132	304	289	298	287
Tuned	549	552	575	243	608	606	539	-

kBájt	tcp_mem	tcp_wmem	tcp_rmem	app_win	wmem_max	rmem_max
Default	38912 51882 77824	4096 16384 1660224	4096 87380 1660224	31	131071	131071
Tuned	65536 131072 16777216	4096 131072 16777216	4096 174760 16777216	15	16777216	16777216

5.2. Az eredmények magyarázata

2. Tézis: A B x D szorzat, az egyidejű TCP kapcsolatok száma, valamint a csomagvesztési arány alapján kidolgozott eljárással, azaz a TCP kernel-változók megfelelő hangolásával jelentős, egyes esetekben közel 200%-os átviteli teljesítménynövekedés érhető el. Ugyanakkor bármely TCP variánst is nézzük, WAN környezetben a torlódásvezérlő algoritmus optimálisra hangolt kernel válozókkal sem képes az 1 Gbit-es kapcsolat fizikai sávszélességéhez közeli átviteli teljesítményt nyújtani még relative alacsony késleltetésű hálózaton sem[C7].

A variánsok közötti különbséget az adja, hogy az alkalmazott AIMD algoritmus milyen dinamikával képes növelni az ablakméretet *(congestion window)*, tehát milyen gyorsasággal kepés alkalmazkodni a rendelkezésre álló sávszélességhez. Ebből a szempontból néhány TCP variáns ablakméret növelő függvénye túl agresszíven növeli az effektív ablakot, ami átviteli teljesítmény szempontjából hatékony megoldás, a másik oldalon viszont ennek hatására súlyosan sérül a párhuzamos adatfolyamokkal szembeni méltányossági *(fairness)* kritérium.

6. TCP vizsgálata védett mobil WiFi hálózatokon

A helyi hálózatok világában a vezetéknélküli átviteli technológiák népszerűsége a mobilitásnak köszönhetően, valamint az alacsony végpontra jutó kiépítési költség miatt dinamikusan növekszik. Ugyanakkor a WiFi átvitel komplex biztonsági problémákat vet fel, melyekre hatékony megoldást kell találnium a hálózatelemzőknek. Éppen ezért különösen fontos megvizsgálni, az új biztonsági megoldásokat (WPA, WPA2) – beleértve az EAP alapú hitelesítési mechanizmusokat és titkosítási protokollokat (TKIP, AES-CCMP) atekintetben, hogy milyen hatást gyakorolnak az IEEE 802.11a/b/g alapú mobil WiFi rendszerek átviteli paramétereire, különös tekintettel az adatkapcsolati szintű roaming időtartamra. Korábbi elemzésekből tudjuk, hogy a mobil kliens fizikai mozgása közben bekövetkező L2-es roaming esemény hatással van mind a TCP, mind az UDP forgalmakra[16, 18, 20].

Amikor az adatkapcsolati rétegben roamin esemény következik be, a mobil terminál lekapcsolódik arról a bázisállomásról, melynek rádiós celláját épp elhagyja, majd megkísérli a kapcsolódást olyan bázisállomáshoz mely új fizikai helyzetében a legelőnyösebb rádiós jellemzőkkel rendelkezik. A következő lényeges lépés az újrahitelesítési fázis, melyet a legújabb EAP mechanizusok már alapszolgáltatásként implementálnak. Így az alábbi kérdések jogosan merül fel bennünk: Az újrahitelesítés mennyivel nyújtja meg a roaming esemény alatti forgalomkiesés idejét. Hogyan hat a vezetéknélküli átviteli közegen bekövetkezett adatvesztés a felsőbb rétegprotokollokra, legfőképp a TCP-re? A szofisztikált adattitkosító mechanizmusok (TKIP, AES-CCMP) dinamikusan cserélik a kliensek unicast kulcsait, mely folyamat extra erőforrást igényel. Továbbá a PMK-t *(Pairwise Master Key)* újra elő kell állítani a roaming esemény bekövetkeztekor. Következésképpen a kulcsgenerálás növeli a roaming időt. A roaming esemény alfolyamatokra bontva (L2 LLC - *Logical Link Control* aktivitás, újrahitelesítés, re-keying) részletesen tudjuk vizsgálni a protokoll többletterheléseket.

6.1. WiFi biztonság áttekintése

WPA

Az IEEE 802.11b szabvány eredeti biztonsági mechanizmusa *(WEP – Wired Equivalent Privacy)* bizonyítottan nem tekinthető biztonságos titkosító megoldásnak [4]. Az IEEE hálózatbiztonsággal foglalkozó osztálya egy magas szintű biztonsági szabvány kidolgozását tűzte ki célul. Így született meg a 802.11i szabvány, melynek célja a 802.11 hálózatok biztonságossá tétele.
A WiFi Alliance egy korai verzióját alkalmazta az említett szabványnak (draft 3.0), kiemelve abból a biztonsági fejlesztések egy olyan részhalmazát, mely képes együttműködni a már létező hardvereszközökkel. Ezt nevezzük WPA *(WiFi Protected Access)* technológiának [3]. A 802.11 szabvány WEP algoritmust definiál a vezeték nélküli hálózatok védelmére. Az eredeti WEP 40 bites RC4 kulcsokat alkalmaz 24 bit inicializációs vektorral (IV), továbbá CRC32 algoritmussal védekezik a csomagbarkácsolás ellen. Azonban ezen algoritmusok mindegyikéről bebizonyosodott, hogy nem elegendőek a megfelelő biztonság eléréséhez. Például az IV hossza túl kicsi, így viszonylag rövid időn belül jó eséllyel újra megjelenthet egy adott érték. Ez a biztonsági hiba nagymértékben megkönnyíti a valósidejű visszafejtést. Továbbá újrajátszás/replay elleni védelmet sem építettek be.

A WPA valójában köztes megoldást ad a vezeték nélküli hálózatokban felmerülő biztonsági kérdésekre. A kulcsok menedzsmentje kétféle mechanizmus alapján történhet: 1. hasonlóan a 802.1x-hez a WPA is támogatja külső autentikációs szerver (pl. RADIUS) és EAP használatát [5]. 2. előre kiosztott (pre-shared) kulcsokkal oldja meg a hitelesítést. Az előbbit WPA-Enterprise-nak, míg az utóbbit WPA-PSK-nak vagy Personal-nak nevezzük. Mindkét mechanizmus master kulcsot generál a kliens *(supplicant)* és a bázisállomás *(authenticator)* számára. A WEP kiváltására TKIP *(Temporary Key Integrity Protocol)* protokollt definiál, mely kompromisszumnak tekinthető a biztonságos kommunikáció és a hardver-kompatibilitás között.

A TKIP RC4 kriptográfiai algoritmust használ a titkosításhoz. Minden csomaghoz saját 128 bites (per-packet) RC4 kulcsot generál. Ezzel megakadályozza a kulcs megszerzésére irányuló *(key recovery)* támadásokat. Emellett beépítettek a WPA-ba újrajátszás *(replay)* elleni védelmet is: a Michael Message Integrity Code algoritmust.

A WPA új, négylépéses Key Handshake algoritmust vezet be a bázisállomás

és a kliens közötti adatforgalom-titkosító kulcsok generálásához és cseréjéhez. Ez a mechanizmus arra is jó, hogy ellenőrizze valóban rendelkezik-e a master kulccsal a bázisállomás és a kliens. A TKIP protokoll további részletes ismertetése a 3. fejezetben olvasható.

WPA2 & IEEE 802.11i

2004 júniusára befejeződött az IEEE 802.11i szabvány hiányzó elemeinek kifejlesztése, melynek hatására a WiFi Alliance létrehozott egy fejlettebb WPA ajánlást, a WPA2-t a 802.11i végleges változata alapján. A WPA2 támogatja a komplexebb és erősebb AES-CCMP *(AES in Counter Mode with CBC-MAC Protocol)* kritogárfiai algoritmust, mely az AES-128 speciális block módú változata. Az erősebb titkosítási követelményeken túl két további előrelépést is tartalmaz a WPA2 a gyors bázisállomások közötti roaming támogatásához. A PMK gyorstárazásával a WPA2 lehetőséget ad a felhasználónak hitelesítés nélküli újrakapcsolódásra egy bázisállomáshoz, melyhez korábban hitelesítve már kapcsolódott. Az előhitelesítés is egy új szolgáltatása a WPA2-nek, mely lehetővé teszi a kliensek előhitelesítését annál a bázisállomásnál, mely felé fizikailag mozog anélkül, hogy lekapcsolódna az aktuális bázisállomásról, melytől távolodik.

6.2. Adattitkosító protokollok

TKIP

A TKIP-t az IEEE 802.11i szabvány definiálja. A protokol tervezőinek egy köztes megoldást kellett találniuk a magas biztonsági szint és a korábbi vezetéknélküli hardverekkel való együttműködés eléréséhez. Ennek megfelelően az új protokoll ugyanazt az RC4-es algoritmust alkalmazza, mint a WEP, ugyanakkor a TKIP-ben működő RC4 128 bites kulcsokkal dolgozik. A legszembetűnőbb változás a csomagonkénti kulcskezelő mechanizmus, melynél minden egyes adatcsomaghoz egyedi kulcs generálódik. Ezek a kulcsok specifikus adatok egy meghatározott köréből állnak elő: küldő fizikai címe (sender MAC address), csomagazonosító (packet ID). Minden egyes csomag tartalmaz egy 48 bites szekvencia számot, mely eggyel növekszik minden egyes csomagküldésnél. A titkosító algoritmus az inicializációs vektor (IV – Initialization Vector) előállításához használja ezt az értéket. Ez a szekvencia

szám garantálja a csomagokhoz tartozó kulcsok egyediségét.

A TKIP 4 darab kulcsból álló készletet határoz meg a kliens és a bázisállomás közötti unicast communikáció számára, valamint további két kulcsot a multicast és a broadcast forgalmak részére [2]. A master kulcsot *(PMK – Pairwise Master Key)* az EAP hitelesítési fázis alatt a hitelesítő szerver (RADIUS) és a kliens közösen állítják elő. Az így létrejött PMK továbbításra kerül a bázisállomás számára egy Access-Accept radius üzenetbe csomagolva. Tovább lépve a bázisállomás 4 utas key handshake-et kezdeményez az ideiglenes kulcsok *(PTK – Pairwise Temporary Key)* generálásához és cseréjéhez. Az algoritmus kiszámítja a PTK kulcsokat, valamint meghatározza a MIC érétket, így ellenőrizve a PMK mindkét oldali (kliens/AP) meglétét.

Az eljárás a következő ideiglenes kulcsokat (PTK - Pairwise Transient Key) használja az unicast adatok titkosítására:

1. Data encryption kulcs: 128 bites kulcs az unicast keretek titkosítására.
2. Data integrity kulcs: 128 bites kulcs, mellyel kiszámolható a MIC az unicast keretekhez.
3. EAPOL-Key encryption kulcs: 128 bites kulcs az EAPOL-Key üzenetek titkosítására.
4. EAPOL-Key integrity kulcs: 128 bites kulcs, mellyel kiszámolható a MIC EAPOL-Key üzenetekhez.

A pairwise ideiglenes kulcsok (PTK) meghatározásához az alábbi értékeket használja a WPA:

1. Pairwise Master Key (PMK): 256 bites kulcs, melyet az EAP-TLS or PEAP hitelesítési eljárásból származtat.
2. Nonce 1: A bázisállomás által meghatározott véletlenszerű érték.
3. MAC 1: A bázisállomás MAC címe.
4. Nonce 2: A kliens által meghatározott véletlenszerű érték.
5. MAC 2: A vezetéknélküli kliens MAC címe.

A PMK-t az EAP hitelesítés során a hitelesítő *(RADIUS)* szerver és a vezeték nélküli kliens együttesen határozzák meg, majd a szerver eljuttatja a bázisállomáshoz egy Access-Accept üzenetben. Ezután az AP kezdeményezi a négylépéses Key Handshake algoritmust:

1. A bázisállomás küld egy Nonce1-et és a MAC1-et tartalmazó EAPOL-Key üzenetet. Mivel az ideiglenes unicast kulcsokat még nem határozták meg, ez az üzenet kódolatlan szöveges formában, integritási védelem nélkül jut el a klienshez. Ezáltal a kliens rendelkezni fog minden információval a pairwise ideiglenes kulcsok kiszámításához.
2. A mobil kliens visszaküld egy EAPOL-Key üzenetet, mely tartalmazza az Nonce2-t és a MAC2-t, továbbá a MIC értéket. A kliens kiszámította az ideiglenes kulcsokat, emellett meghatározza a MIC értéket is, felhasználva EAPOL-Key integrity kulcsot. A bázisállomás Nonce 2 and MAC 2 értékek alapján határozza meg az ideiglenes kulcsokat, és érvényesíti a MIC értékét.
3. Az AP ismételten küld egy EAPOL-Key üzenetet egy MIC értékkel és egy kezdeti szekvencia számmal. Ezzel jelzi, hogy készen áll titkosított unicast és EAPOL-Key üzenetek küldésére.
4. A mobil kliens visszaküld egy EAPOL-Key üzenetet egy MIC értékkel és egy kezdeti szekvencia számmal. Ezzel jelzi, hogy ő is felkészült titkosított unicast és EAPOL-Key üzenetek küldésére.

A fenti lépések kettős célt szolgálnak: 1. az ideiglenes kulcsok (PTK) meghatározása. 2. a kapott MIC értékkel ellenőrzhető, hogy mind a kliens, mind a bázisállomás ténylegesen rendelkezik-e a PMK-val.

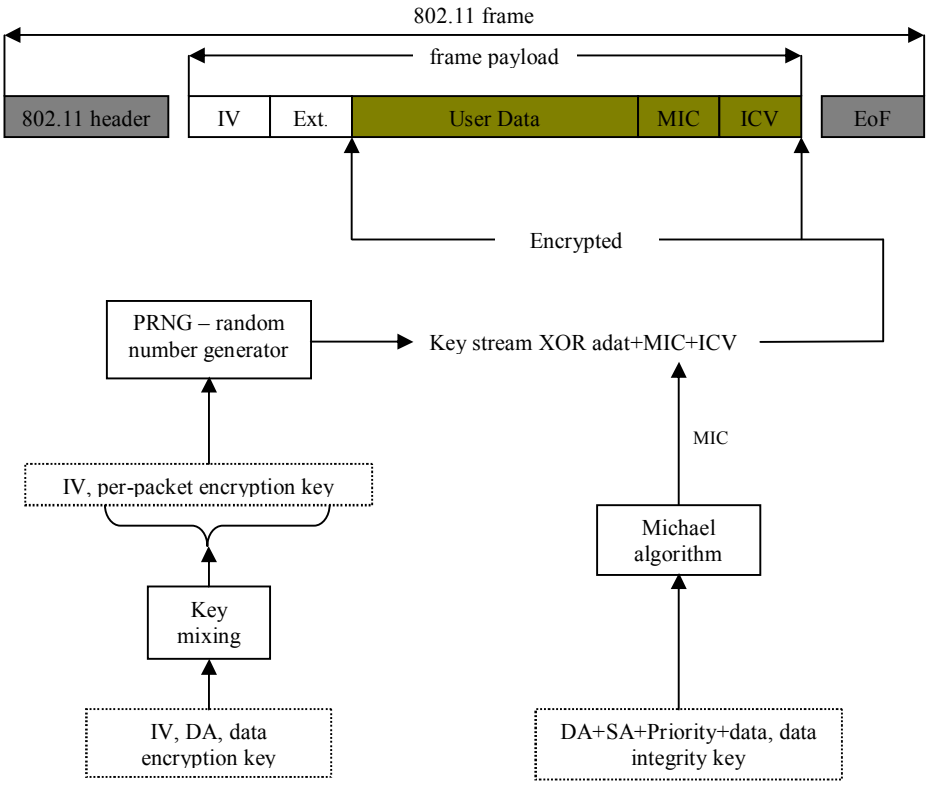
Az alábbi komponensek szükségesek az adatkeretek titkosításához:

- Inicializációs Vektor (IV);
- A PTK adat- vagy csoporttitkosító kulcsa;
- A keret forrás és cél fizikai címe (DA, SA);
- A prioritás mező értéke, mely alapértelmezésben ’0’;
- A PTK adatintegritás vagy csoportintegritás kulcsa.

TKIP titkosító algoritmus:

1) A WPA keverő függvényehez *(mixing function)*, mely a csomagonkénti kulcsokat számítja ki az IV, DA és Data Encryption Key szolgáltatja a bemenetet.

2) A MIC érték előállításához a Michael adatintegritási algoritmus bemenetei a következők: DA, SA, prioritás érték, titkosítás nélküli adat és az adatintegritási kulcs.
3) Az ICV-t a CRC-32 checksum-ból kerül meghatározásra.
4) Az RC4 programozott véletlenszám generátora az alábbi bemeneteket használja a kulcsfolyam előállításához: az IV és a csomagonkénti egyedi kulcs. A kulcs folyam mérete megegyezik az adatmező a MIC és az ICV értékek együttes méretével.
5) A titkosított 802.11 adat előállításához végrehajt egy logikai XOR műveletet a kulcsfolyam az adapt, a MIC és az ICV kombinációján.
6) Végezetül hozzáadja az IV-t a titkosított adathoz és becsomagolja az eredményt egy 802.11 fejrész és egy EFF *(End of Frame Filed)* közé.



10. ábra A TKIP blokk sémája

AES-CCMP

A CCMP hasonlóan a TKIP-hez ideiglenes kulcsokat (PTK) alkalmaz az adatok titkosítására. A PTK meghatározásahoz a TKIP-nél már ismertetett 4 lépéses Key Handshake eljárást használja. Mivel tartalmaz adatintegritási védelmet, ezért egyszerre válthatja ki a TKIP-t és a Michael algoritmust. Az AES-CCMP a 802.11 payload és MIC titkosításához counter módú AES-t alkamaz, és a MIC meghatározását CBC-MAC algoritmussal végzi, az alábbiak szerint:

1) AES-sel titkosít egy 128bites kezdő blokkot és a data integrity kulcsot. Ez 128 bit hosszúságú kódot eredményez (X1).
2) Végrehajt egy XOR műveletet az előbbi 128 bites kódón és a következő 128 bites adatblokkon, miközben a MIC értéket is meghatározza. Az éredmény szintén egy 128 bites kód (X2).
3) Az X2 kódot titkosítja AES-sel a data integrity kulcsot felhasználva. Így létrejön X3.
4) Ismét XOR műveletet hajt verge X3-n és a következő 128 bites adatblokkon.

Az eljárás a 3. és 4. lépést ismétli minden újabb 128 bites adatblokkra. A 128 bites kód felső 64 bitje lesz a MIC érték.

6.3. Mérési környezet és a mért adatok ismertetése

A mérésekhez alkalmazott eszközök mindegyike képes IEEE 802.11a/b/g szabványoknak megfelelő kommunikációra, valamint támogatja az általunk tesztelt biztonsági technológiákat. Vizsgálatunk fókuszában a különböző biztonsági mechanizmusok és rádiós átviteli technológiák mozgó kliensre kifejtett együttes hatása állt. A mozgó kliens cellaváltásakor bekövetkező roaming folyamatot befolyásoló hatásokat mértük és elemeztük, ugyanis a roaming idő alatt kieső forgalom mértéke jelentősen befolyásolja a hálózati alkalmazások szolgáltatási minőségét. A mobil kliens 5-6 km/h (1,4-1,7 m/sec) sebességgel haladt a bázisállomásokat összekötő egyenessel párhuzamos irányban oda-vissza. Egy mérési periódus (TSi) alatt az MT L2 roaming hatására az AP1-ről az AP2-re asszociált, majd visszafelé haladva újabb L2 roaming hatására visszakerült az AP1 hatáskörébe.

A mobil terminál egy notebook volt, amelyen FTP kliens futott. A TCP kapcsolat huzalos végén egy Linux alapú csomópontot helyeztünk el, melyen FTP szerver futott. Nagyméretű állományt mozgattunk a kliens és a szerver között. Az átviteli sebességet 256Kbyte/sec-re korlátoztuk, ami jól közelíti egy átlagosan (10-15 klienssel) terhelt bázisállomáson mérhető effektív felhasználói sávszélességet. Fájl letöltésre és feltöltésre is elvégeztük a méréseket, ugyanis a bázisállomás más-más viselkedést mutat a forgalom irányának megfelelően.

Mindkét bázisállomás ugyanabban az L2-es VLAN-ban helyezkedett el, fizikailag egymástól 50 méteres távolságban. Vezetékes oldalon a bázisállomások forgalmát tükröztük egy monitor/Span VLAN-ba, ahol a kereteket egy Linux-os munkaállomáson tcpdump segédprogrammal kaptuk el, és mentettük le. A mért adatok elemzéséhez az Ethereal 10.0.14-es verzióját használtuk. A bázisállomások által kisugárzott rádiós teljesítményt 802.11b/g esetén 5mW-ra, míg 802.11a esetén 12mW-ra állítottuk be, figyelembe véve a méréshez szükséges optimális cellaméretet. Az említett IEEE 802.11a/b/g rádiós szabványokra először nyitott (open), hitelesítés nélküli kommunikációval végeztük el a méréseket, majd PEAPv0, v1 hitelesítéssel (PEAP-GTC *Generic Token Card,* PEAP-MSCHAP *Microsoft Challenge Handshake Authentication Protocol*) és titkosítással (TKIP, CCMP) az 1. táblázatban megadott protokoll-kombinációkat alkalmazva:

Technológia	*Hitelesítés*	*Titkosítás*
WPA	PEAP-MSCHAPv2	TKIP
	PEAP-GTC	TKIP
WPA2	PEAP-MSCHAPv2	AES-CCMP
	PEAP-GTC	AES-CCMP

1.táblázat Biztonsági megoldások

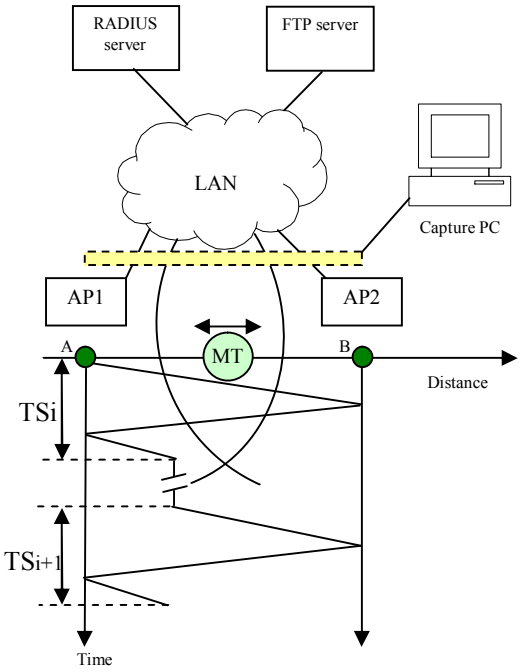
Az MT (mobil terminál) az A pontból haladt a B pontba, majd vissza. Az A és B pont közötti távolság 50 méter. A biztonsági technológiák roaming időtartamra, valamint TCP dinamikájára gyakorolt hatásának vizsgálatához az alábbi időpillanatokat határoztuk meg. A cellaváltás időtartamát jelentősen befolyásolja a bázisállomások beacon periódusa. Korábbi vizsgálatainkból kiderült [1], hogy IEEE 802.11g/b esetén 50ms alatti periódusidővel érhető el a

legkedvezőbb cellaváltási idő. Ennek megfelelően méréseinket 802.11b/g esetében 40ms-os beacon periódussal vegéztük. IEEE 802.11a esetén 50ms körüli értékkel kapunk elfogadható roaming teljesítményt. Ezért itt 50ms-ra állítottuk be a periódust. Mindemellett érdemes megjegyezni, hogy a túl alacsony baecon periódus többszörös roaming esemény idézhet elő a A pontból B pontba történő haladás közben az épületfalak reflexiós hatásának köszönhetően.

Momentumok	*Eseményleírás*
T1	L2 roaming előtt az MT által küldött utolsó TCP csomag (LIP)
T2	Újrahitelesítés kezdete
T3	Újrahitelesítés vége
T4	L2 roaming után az MT által küldött első TCP csomag (FIP)

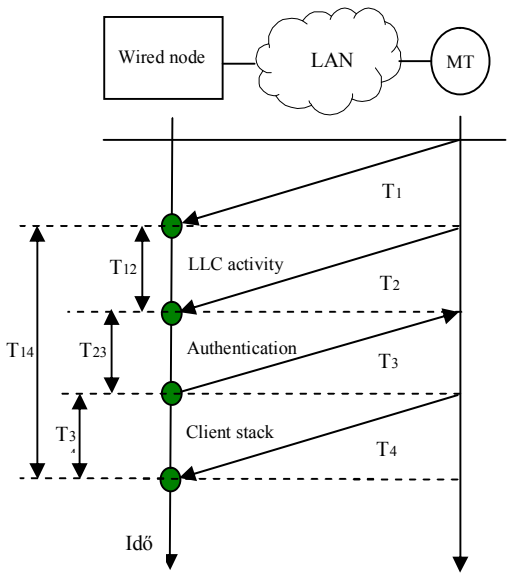
2. táblázat Jelölések

Amint azt a 2. táblázat is mutatja a fontos TCP csomagok (LIP, FIP) az FTP session-höz tartoznak, ezért fontosnak tartottuk megkülönböztetni őket az egyéb TCP forglamaktól, melyeket a kliens generált a capture periódus alatt. Az első, T_{12} időtartam a L2-es roaming folyamat időtartama, ennek részleteit korábbi cikkünkben mutattuk be[1]. A T_{23} újrahitelesítési időtartam alatt végbemegy a 802.1x autentikáció a kliens és a RADIUS szerver között, meghatározzák a PMK-t, majd a bázisállomás és a kliens végrehajtja a 4 lépéses Key Handshake algortimust, mely az ideiglenes kulcsokat állítja elő az adatforgalom titkosításához.



11. ábra Tesztrendszer

A $T_{1...3}$ fázisokat követően újraindulhat a titkosított adatforgalom a kliens és a bázisállomás között. A cellaváltás alatt a legtöbb esetben nem szakad le a TCP kapcsolat, viszont jelentős forgalomkiesést jelentkezhet adatkapcsolati szinten. A T_{34} érték jól mutatja, hogy a TCP kapcsolat nem azonnal éled fel, az újrahitelesítés és az első hasznos TCP csomag között jelentős időkülönbségek adódhatnak, melynek mértéke a végpontokon alkalmazott TCP variáns(-ok) függvénye.

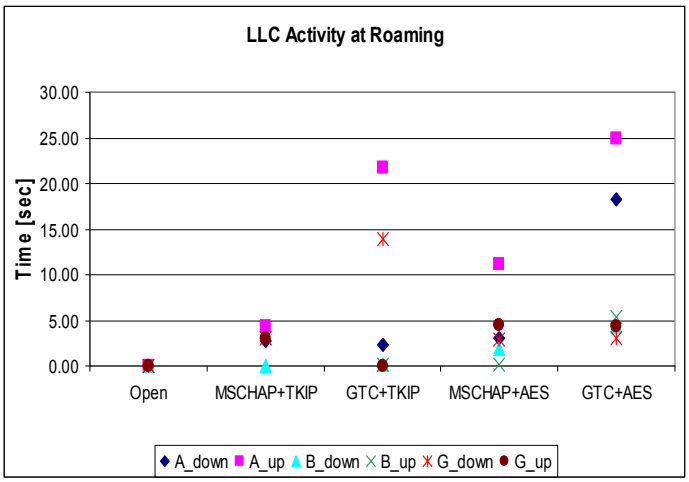


12.ábra Mérési periódusok

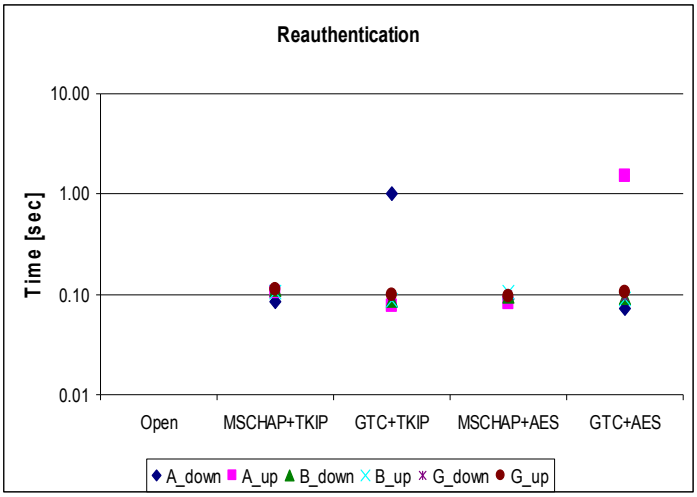
6.4. Új eredmények

A méréskor a vezetékes oldal teljes adatforgalmát lementettük állományokba, így elemzéskor a roaming folyamatot jellemző jelenségekről teljesebb képet kaptunk. Az L2-es roaming folyamat során a mobil kliens jól meghatározható keretsorozatot küld az új bázisállomásnak, így minden capture állományban egyszerűen meg tudtuk állapítani a roaming esemény időpontját. Mivel TCP forgalmat vizsgáltunk, így természetesen a legfontosabb kérdés az volt, hogy milyen hatásai lesznek a cellaváltás során végbemenő roaming alfolyamatoknak (L2 LLC aktivitás, újrahitelesítés, re-keying) a TCP adatfolyamra. Továbbá lényeges, hogy az eltérő 802.11-es rádiós technológiák milyen időtartambeli különbségeket mutatnak. Az alábbi grafikonokon a roaming folyamatot négy fázisra bontva mutatjuk be, hangsúlyozva a fogalom irányának fontosságát. Bármely részfolyamatot vizsgáljuk, jelentős különbségek figyelhetők meg a hitelesítési mechanizmusok (MSCHAP és GTC) és az egyes rádiós technológiák (IEEE 802.11a/b/g) között is. Az LLC aktivitás időtartama nagymértékben függ az alkalmazott PEAP típustól (4. ábra). Itt az MSCHAP minden esetben gyorsabb L2 roamingot eredményezett. Érdemes megfigyelni azt is, hogy az időtartam a TCP forgalom irányának is függvénye. Ebben a szakaszban az L2-es roaming minden esetben gyorsabb volt MS-CHAP-ot alkalmazva. Természetesen a roaming intervallum a TCP folyam irányának is függvénye[J3].

Ha a roaming akkor következik be, amikor a kliens fájlletöltést hajt végre, abban az esetben előfordulhat, hogy néhány csomag már beérkezett annak a bázisállomásnak pufferébe, melytől távolodik a kliens és már megkezdte a kapcsolódást az új bázisállomáshoz. Ezeket a csomagokat újra kell küldeni az új bázisállomás irányába közvetlenül azután, hogy az L2-es switch frissítette a CAM tablájában a mobil kliens MAC-címét. Következésképpen minél hosszabb a roaming periódus, annál több csomagot kell újraküldeni, ami rontja a TCP átviteli teljesítményét, valamint a hálózati alkalmazások számára biztosított szolgáltatás-minőséget (QoS). A következő grafikonokon (4-6. ábra) a 802.11 kommunikáiókat (a 802.11 típusnak megfelelő) nagybetűvel jelöltük és egy utótag (_up and _down) jelzi a forgalom irányát a mobil kliens szemszögéből.



13. ábra L2 Roaming period



14. ábra Re-authentication period

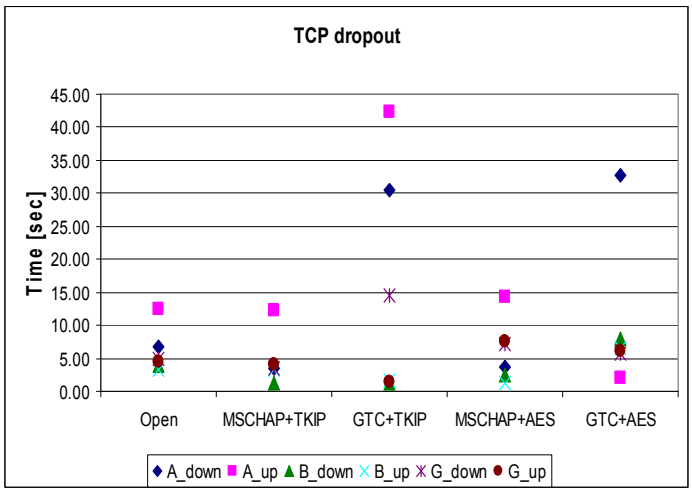
Az újrahitelesítési szakaszban jóval kisebb (<20ms) az eltérés a hitelesítési mechanizmusok között (5. ábra). Minden hitelesítési időtartam 100ms körül ingadozik. A grafikonon látható extrém eltérések sokkal inkább a rádiós technológiák közötti különbségekre világítanak rá. Az IEEE 802.11a szabvány minden mérési intervallumban gyengébben teljesített a 802.11b/g technológiákhoz képest. Szélsőséges esetekben (802.11a+GTC+TKIP, 802.11a+GTC+AES) az újrahitelesítés időtartama (~1000ms) egy nagyságrenddel nagyobb az átlagosan mért 100ms-os értéknél. A TCP forgalomkiesés irányérzékenységének oka az, hogy a bázisállomás pufferelést végez a beérkező csomagokra. Mivel két jelentősen eltérő viselkedésű hálózattípus kapcsol össze, ezért a

A teljes TCP forgalomkiesést a 6. ábra szemlélteti. A grafikon a kliens által a régi bázisállomásnak küldött utolsó és az új bázisállomás irányába küldött első hasznos TCP csomag közötti intervallumokat mutatja. A legkedvezőtlenebb értékek a 802.11a és GTC hitelesítés kombinációjával adódtak. Viszont az is látszik, hogy különösen jó eredmény érhető el 802.11g/b és PEAP-MSCHAP együttes alkalmazásával, egyes esetekben (MSCHAP+TKIP, MSCHAP+AES) az open autentikációt megközelítő értékek adódtak a roaming folyamat időtartamára.

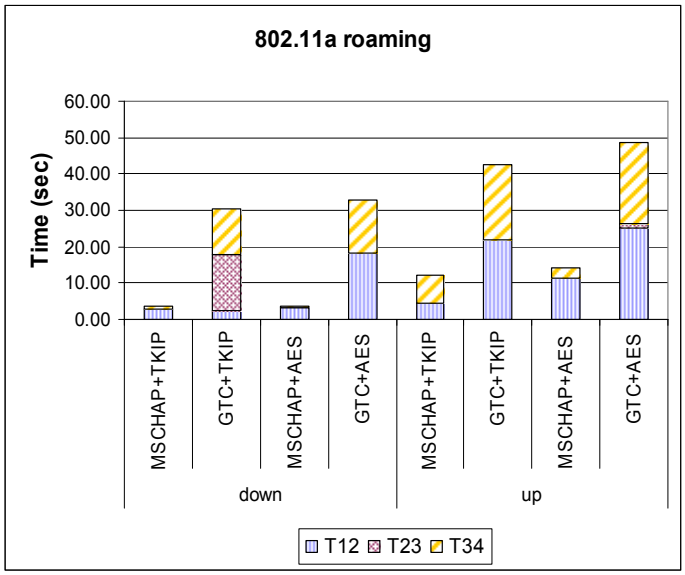
Az újrahitelesítési fázisban (5. ábra) a hitelesítési mechanizmusok közötti

különbség sokkal kisebb nagyságrendű (<20ms). Mindegyik hitelesítési periódus 200ms körül mozog. Az ábrán látható extrém eltérések a 802.11-es technológiák közötti különbséget mutatják. Bármely mérési fázist nézzük, az IEEE 802.11a adta a legkedvezőtlenebb roaming teljesítményt a 802.11b/g-hez viszonyítva. Szélsőséges esetekben (802.11a+GTC+TKIP, 802.11a+GTC+AES) az újrahitelesítés ideje (~1000ms) egy nagyságrenddel nagyobb, mint a 100ms mérési átlag. A TCP irányérzékenységének oka, hogy a bázisállomás puffereli a bejövő csomagokat, amely a csomagok újraküldését okozhatja az új bázisállomás felé a roaming esemény befejezése után. Ezen kívül a bázisállomás két alapjaiban különböző átviteli médiumot kapcsol össze.

A teljes TCP kiesést a 6. ábra mutatja. Ez a grafikon a kliens által a korábbi bázisállomásnak küldött utolsó fontos csomag (LIP) és az új bázisállomásnak küldött első fontos TCP csomag (FIP) közötti intervallumokat mutatja be.. A legkedvezőtlenebb értékek 802.11a és PEAP-GTC kombinációjával adódtak. Ugyanakkor különösen jó eredményeket értünk el 802.11g/b és PEAP-MSCHAP kombinációjával, amikor néhány esetben (MSCHAP+TKIP, MSCHAP+AES) a mért értékek - a PMK cache funkcióval - megközelítették a nyitott hitelesítés értékeit.



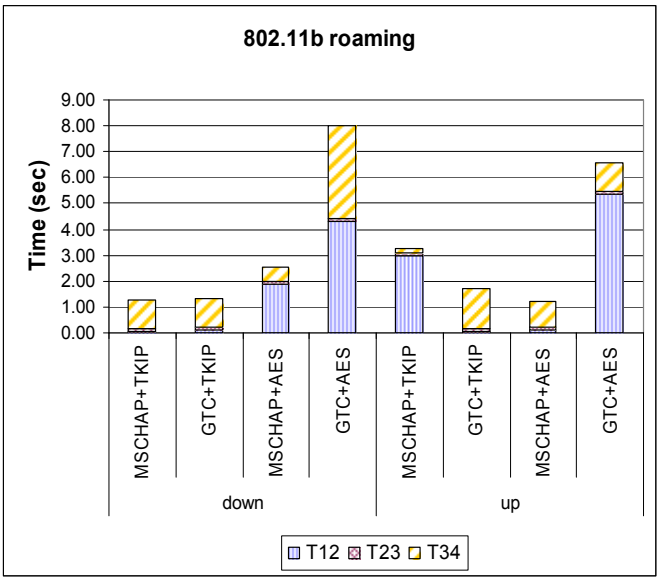
15. ábra Teljes TCP kiesés



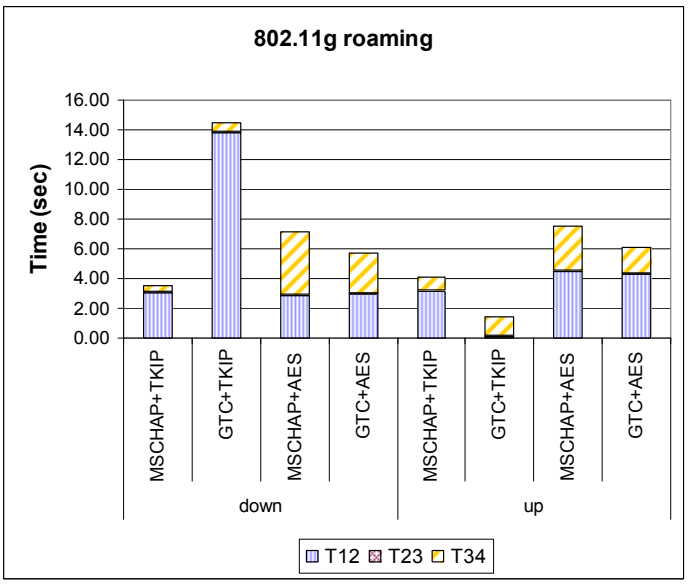
16. ábra Hitelesítés és titkosítás: 802.11a

Ha összevetjük az IEEE 802.11 rádiós technológiák roaming teljesítményét (7., 8., 9. ábra), azt tapasztaljuk, hogy a legmagasabb értékek 802.11a szabvány esetén jelentkeznek. Ennek magyarázata az, hogy a mikro cellák mérete ennél a technológiánál kisebb, mint az azonos sugárzási teljesítményű 802.11b/g állomások esetén. Így az 50 méteres távolságban levő bázisállomások cellái kevésbé vannak átfedésben. Ennek ellensúlyozására növelhetjük a kisugárzott rádiós teljesítmény, ekkor viszont ügyelni kell arra, hogy az 5,4GHz-es tartományban üzemelő 802.11a szabvány jóval alacsonyabb maximális adóteljesítményt tesz lehetővé (40mW).

Amikor összehasonlítjuk a 802.11 technilógiák roamin paramétereit (7., 8. és 9. ábra), akkor észrevesszük, hogy a 802.11a IEEE szabvány nyújtja a legmagasabb, egyben legkedvezőtlenebb értékeket. Ezzel a technológiával azonos kimenőteljesítmény esetén kisebb fizikai méretű mikorcellák alakul ki, mint a 802.11b/g esetén. Következésképpen két 802.11a bázisállomás egymástól 50 méteres távonságból kevésbé átfedő cellákat hoz létre. A jelenség kompenzálásához növelhetjük a kisugárzott rádiós teljesítményt. Ugyanakkor számításba kell venni az 5,4 Ghz-es tartományban működő 802.11a számára meghatározott 40mW teljesítménykorlátot és beltéri sugárzás esetén a falak hullámvisszaverő hatását[J3].



17. ábra Hitelesítés és titkosítás: 802.11b



18. ábra Hitelesítés és titkosítás: 802.11g

6.5. Az eredmények magyarázata

Ebben a fejezetben megvizsgáltuk az IEEE802.11i alapú biztonsági megoldások teljesítményét IEEE802.11a/b/g mobil vezetéknélküli rendszerekben. Hangsúlyozottan az adatkapcsolati szintű roaming eseményre fokuszáltunk. Azt láttuk, hogy a 802.11a szabvány mobilitás tekintetében nem igazolta hatékonyságát. Nyilvánvalóan a hosszabb mérési intervallumokon túl a roaming esemény a TCP kapcsolatok lebontását is eredményezte számos esetben, annak ellenére, hogy a 802.11a a kevésbe terhelt, tehát kevésbe zajos 5.4GHz-es tartományban üzemel. Mindegyik biztonsági protokoll kombináció többletterhelést jelent a roaming időre. Következésképpen az alkalmazási rétegben érzékelhető QoS paraméterek minőségét is csökkentik. Bármely roaming szakaszt is nézzük, jelentős különbségek adódnak a hitelesítési mechanizmusok (MSCHAP, GTC) és a vezetéknélküli technológiák között is (IEEE 802.11a/b/g). Az LLC (Logical Link Control) aktivitás ideje valójában az alkalmazott PEAP verziótól (vagyis a tunel-en belül használt hitelesítéstől) függ. Ebben a tekintetben az L2-es roaming minden esetben gyorsabban ment végbe MSCHAP belső hitelesítéssel. Ennek ellenére a hitelesítő és titkosító mechanizmusok közötti eltérések jóval kisebb nagyságrendűek, mint a szállítási rétegben tapasztalható késleltetés, mely nagyságrendileg a másodperces tartományban mozog. Az újrahitelesítési fázisban szintén az tapasztaltuk, hogy jóval kisebb mértékű az egyes EAP típusok közötti eltérés, ami leginkább az egyes protokollok komplexitásának, a feldolgozási időnek és az üzenetváltások számának függvénye[J3].

A mobil WiFi környezetben elérhető alkalmazási rétegbeli szolgáltatás minőség a komplex roaming folyamattól függ, ahol az összteljesítmény a protokoll stack számos komponensének függvénye, hangsúlyozva az alkalmazott TCP torlódásvezérlő mechanizmus jelentőségét. Minél hosszabb a roaming periódus, annál nagyobb számú TCP csomagot kell újraküldeni, ami ennek megfelelően rontja a TCP átviteli teljesítményét és a szolgáltatásminőséget.

7. TCP vizsgálata IPv6 mobil WiFi környezetben

7.1. Bevezetés

Az Internet2 legjelentősebb előnyeként említhető az IPv6 feletti szofisztikált mobil szolgáltatások megjelenése és elterjedése[15]. Izgalmas felhasználói és szakmai kérdéskörnek fogalmazódik meg a mobilitás hatásának mértéke a TCPv6, UDPv6 protokollokra épülő szolgáltatások viszonylatában. Ahhoz, hogy ebben a témában a minőségi választ mennyiségi jellemzőkkel is érzékeltethető legyen, szükséges olyan összehasonlító mérések elvézése, amelyek előtérbe helyezik az IPv4 és IPv6 technológiák közötti különbségeket mobil környezetben. A szolgáltatások működőképességének biztosítása a csomópont fizikai mozgása közben olyan igény, amely a korszerű hálózatok igény-palettáján joggal jelenik meg. A vezeték nélküli IP telefónia, a vezeték nélküli laptop és a PDA számítógépek fejlődése erőteljesen ebbe az irányba mutat.

A vezeték nélküli LAN-ok mobilitás tulajdonsága újabb hasznos eredménykhez vezet:

i) *innovatív alkalmazás fejlesztés*: vészjelzések, üzenetek küldése, folyamatos hálózati kapcsolatban álló munkafolyam rendszerek megjelenése;

ii) *hatékonyság és termelékenység növekedés*: a folyamatos hálózati kapcsolódás lehetővé teszi a munka bárhonnani elvégzését időkiesés nélkül;

iii) *adatok megnövelt hitelessége*: az adatok bármikor és bárhonnan elérhetők;

iv) *rendelkezésre állás*: a felhasználó virtuálisan online kapcsolatban maradhat az otthonában, az utcán és a munkahelyén is.

Jelen anyagban a mobil WiFi hálózaton kommunikáló IP terminálok különböző alkalmazásainak mennyiségi összehasonlítását mutatjuk be, magyarázatot adunk a tapasztalt jelenségekre és következtetéseket vonunk le a várható fejlesztési igényekre vonatkozóan.

7.2. Mobil adatátvitel

Mint ismeretes, az IP protokoll 4-es és 6-os veziója a hagyományos rögzített, huzalos hálózati kommunikáción túlmenően képes ellátni mobil funkciókat is[15]. A vezeték nélküli adatátviteli kapcsolatok a hálózati réteg számára ugyancsak képesek biztosítani a keretek továbbítását[13]. Ezáltal az IP protokoll verziójától függően a szállítási rétegben működő szolgálatok részére kisebb vagy nagyobb mértékben történik az alsóbb rétegek viselkedésének figyelembe vétele. Az IP protokoll mobil funkciója arra vonatkozik, hogy a terminál kommunikáció közben elmozdul fizikai helyéről, minek következtében megváltozik a hármas rétegbeli hálózati környezete. Ehhez a meglátogatott új helyszínen egy "foreign agent" funkció betöltését biztosító útválasztó egy IP feletti IP alagút segítségével továbbra is kapcsolatban áll a „home agent", eredeti útválasztóval [19].
Ezáltal a mozgó terminál az új helyszínen továbbra is képes kommunikálni.

Fontos kérdés, hogy az "agent" processzek közötti interakció mennyire gyors, illetve milyen további terhelést okoz a hálózaton. Az IP protokoll 4-es és 6-os verziója ebből az aspektusból is lényegesen eltérő viselkedést mutat. Ezen tulajdonságokat a 3. táblázatban soroljuk fel:

Jellegzetes tulajdonságok	**Mobil IPv4**	**Mobil IPv6**
Speciális routing funkció (foreign agent)	Igen	Nem
Útvonal optimalizációs képesség	A protokoll része	Bővítmény
Szimmertikus kapcsolat az MT és a router között Symmetrical connection between the MT and the router in the current location	Nem	Igen
A routing többletterhelés sávszélesség igénye	Magas	Alacsony
Az adatkapcsolati rétegről való lekapcsolódás képessége	Nem	Igen
"Tunnel soft" állapot kezelése szükséges-e	Igen	Nem
"Dynamic home agent" cím felfedezése	Nem	Igen

3. táblázat

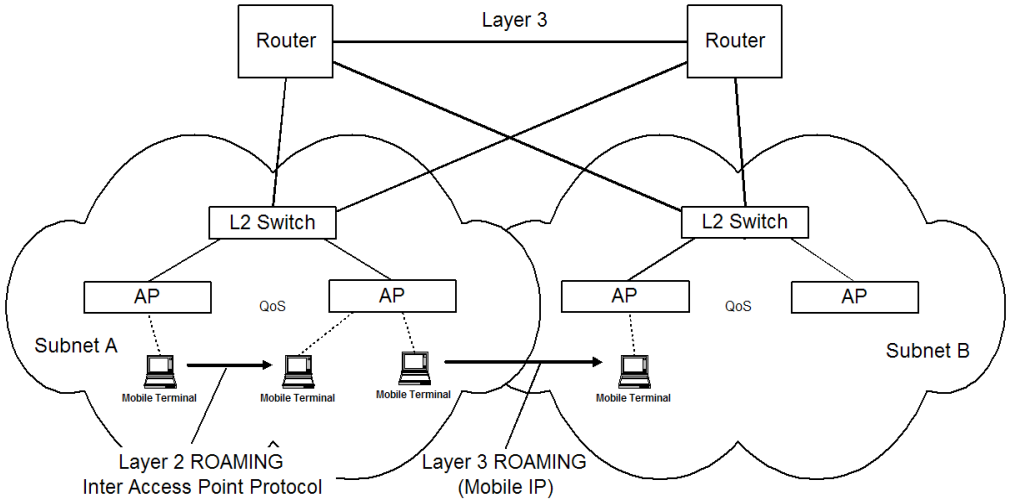
A vezeték nélküli adatkapcsolatok esetén lehetőség van arra is, hogy a mobil terminal ugyanabban az ütközési tartományban maradjon, vagyis a terminál IP címének változása nélkül módosuljon a forgalmazott keretek útvonala. Ez tipikusan az L2 roaming esete, amikor a mobil terminál úgy vált bázisállomást, hogy csak az adatkapcsolati eszközök CAM táblájának tartalma módosul. A fejezet valós kültéri mobil WiFi rendszer környezetben egy mozgó járműben elhelyezett terminál által generált rádiós cellaváltás alatt bekövetkező roaming folyamat IPv4, illetve IPv6 kapcsolatokra kifejtett hatását vizsgálja meg.

Hálózat	**Protokoll**	
	IPv4 / IPv6	Mobil IPv4 / Mobil IPv6
Huzalos	√	√
Vezetéknélküli	√	√

4. táblázat

7.3. Roaming mechanizmusok

A vezeték nélküli helyi hálózatok lehetővé teszik, hogy a csomópontok a vállalati hálózathoz virtuálisan kapcsolódjanak. A cellaváltás *(roaming)* olyan időben lejátszódó folyamat, amely során a mobil terminál egyik kiszolgáló AP-bázisállomástól egy másik AP-ra csatlakozik rá. Adatkapcsolati (L2) roamingról beszélünk, ha a folyamat azonos IP alhálózatba tartozó AP-k között történik (1. ábra.).



19. ábra L2-es és L3-as roaming

Ha a terminál másik alhálózatba tartozó új AP-hoz csatlakozik, akkor hálózati (L3) roamingról beszélünk. Hálózati cellaváltás az adatkapcsolati roaming sikeres lezajlása után következhet be[19]. A cellaváltás mindig a terminál döntésén alapul, amelynek feladata a lehetséges bázisállomások felderítése, az ezekhez tartozó paraméterek értékelése, majd a szóbajövő cellák közül az új kiválasztásának eldöntése.

Az adatkapcsolati cellaváltás az alábbi fázisokat foglalja magába:

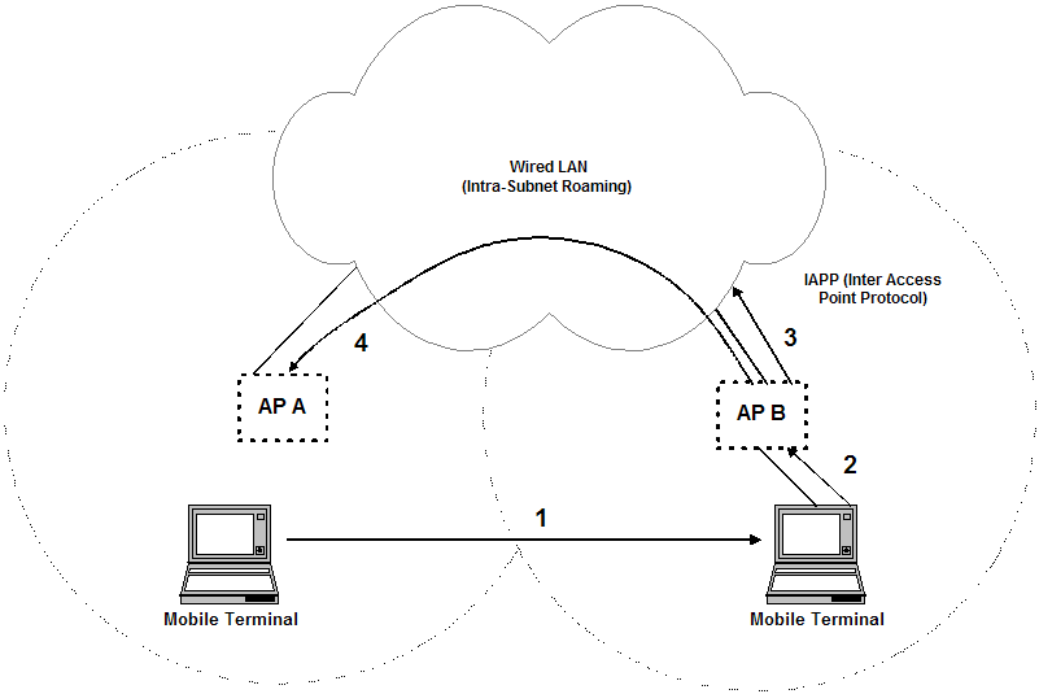
1. A terminál az “A” cellából elmozdul a “B” cellába. A bázisállomások ugyanabban az
alhálózatban vannak, így L2 roamingról beszélünk. Ahogy a terminál kilép az “A” cellából, az APA bázisállomással fennálló kapcsolat paraméterei közül valamelyik átlépi a megadott küszöb értéket, s ez kiváltja a roaming folyamat indítását.

2. A kliens végig elemzi az összes IEEE 802.11-es csatornát, lehetséges bázisállomást keresve. Megtalálja az APB-t, lezajlik a fizikai rádiós csatornán a hitelesítés és az asszociáció folyamata.

3. Az APB a kliens alhálózatába egy nulla tartalmú multicast üzenetet küld, amelynek forrás fizikai címe éppen a mobil terminál címével egyezik meg.

Ez alapján a huzalos LAN hálózatban található switch-ek frissítik kapcsolási táblájukat. Így a terminálnak címzett Ethernet keretek ezután nem az APA, hanem az APB bázisállomáshoz kerülnek.

4. Az APB a saját forrás MAC címével küld egy multicast üzenetet, amelyben értesíti az alhálózat összes bázisállomását arról, hogy az adott MAC című terminál hozzá asszociált. Ahogy az APA ezt megkapja, törli a mobil terminál MAC címét az asszociációs táblájából.



20. ábra. Az L2-es roaming lépései

A roaming folyamatot mindig a kliens kezdeményezi, de a folyamatra vonatkozóan még nem létezik IEEE szabvány. A Cisco gyártmányú terminálok esetében az alábbi események váltják ki a roaming folyamat indítását:

a.) *Maximális csomagküldés próbálkozási szám átlépése.* Ha a kliens a maximum data retry–ként megadott számú próbálkozás után sem tudja a csomagot elküldeni, elindítja a roaming folyamatot. A Cisco Aironet kliensben ez az érték alapértelmezés szerint 16, és az Aironet Client Utilityben állítható.

b.) *Túl sok "beacon" kihagyása.* Minden, bázisállomáshoz társított kliensgép periodikusan kap „beacon" keretet. Alapértelmezésben 100 milliszekundumonként küld „beacon"-t a bázisállomás. Ez a periódus egyben konfigurációs paraméter is. A terminál a „beacon"-ben található érték alapján megtanulja annak periódusát. Amennyiben a terminál nyolc periódus ideig nem kap „beacon"-t, kezdetét veszi a roaming folyamat. A beérkező „beacon"-ök folyamatos figyelésével - még egy „idle" állapotban levő kliens is - képes érzékelni a vezeték nélküli kapcsolat minőségének romlását, majd pedig roaming-ot kezdeményezni.

c.) *Átviteli ráta váltása.* Normál esetben a rádiós keretek átvitele a bázisállomás alapértelmezett adatátviteli sebességével történik. Ez a ráta a legmagasabb átviteli sebesség, amelyet „required" vagy „enable" paraméterként lehet az AP-n beállítani. Minden olyan alkalommal, amikor egy csomagot alacsonyabb sebességgel kell újraküldeni, a "retransmit" számláló hárommal növekszik.
Minden olyan csomag esetében, amikor az alapértelmezett átviteli sebességgel sikerült a
továbbítás, ez a számláló eggyel csökken egészen addig, amíg a nulla értéket el nem éri.

Amennyiben a számláló eléri a 12-es felső határt, az alábbi események valamelyike következik be:

- ha a kliens nem hajtott végre cellaváltást az elmúlt 30 másodpercben, akkor bekövetkezik a gyors cellaváltás (fast roaming);
- ha az említett időn belül *roaming*-ot hajtott vége, akkor eggyel alacsonyabb fokozatra csökkenti az átviteli rátát.

Az alapértelmezett átvitelnél alacsonyabb rátájú sikeres átvitel esetén, egy rövid idő elteltével ismét visszaugrik az eggyel magasabb sebességű üzemmódba.

d.) *Periódikus kliens intervallum (opcionális).* A Cisco Aironet v6.1-től kezdve konfigurálni lehet, hogy a mobil terminál milyen gyakorisággal, illetve milyen jelerősség mellett keressen jobb vételi minőségű bázisállomást. Ezekkel a beállításokkal a terminál egy jobb térerejű bázisállomást fog keresni feltéve, hogy az alábbi feltételek mindegyike teljesül:

- A terminál már legalább 20 másodperce asszociált az aktuális AP-hoz. Ez a feltétel megakadályozza, hogy a kliens túl gyorsan

kapcsoljon a bázisállomások között. Érvényes értékek 5-255 másodperc.

- A térerősség 50%-nál gyengébb. Érvényes intervallum: 0-75%-ig.

e.) *Kliens inicializáció.* A terminál bekapcsolásakor és újraindításakor lezajló folyamat.
A roaming folyamathoz új bázisállomás keresése szükséges[19, 24]. Ennek érdekében a terminál a rádiós csatornákon scan technika segítségével meghatározza az elérhető bázisállomások listáját, amelyből a legjobbat választja ki. A scan technika csatornánként egy-egy „probe" teszt üzenet küldését jelenti, amire „probe" válasz vagy „beacon" érkezik a csatornán üzemelő bázisállomástól. Az AP-tól érkező „beacon"-öket csak akkor veszi figyelembe a kliens, ha az SSID és a titkosítási beállítások megegyeznek.

A keresés befejezése után a listából kiválaszt egy bázisállomást, hogy az elérési paramétereit összehasonlítsa a lista többi tagjával. Ha a terminál kezdeti „start-up" fázisban van, akkor az új AP a listában elsőként szereplő tag lesz; ha a terminál roaming fázisban van, akkor az új AP a korábbi marad amennyiben válaszolt a teszt „probe" keretekre. Válasz hiánya esetén a lista első tagja lesz az új AP.

Az aktuális AP a lista többi elemével összehasonlításra kerül. Ahhoz, hogy egy tag új AP lehessen, minden listabeli AP-nak az alábbi szempontokak kell teljesítenie:

1) A potenciális cél AP jelerőssége legalább 20%. Ha a térerő több mint 20%-kal gyengébb, mint az aktuális AP térereje, akkor legalább 50% jelerősséggel kell rendelkezzen.

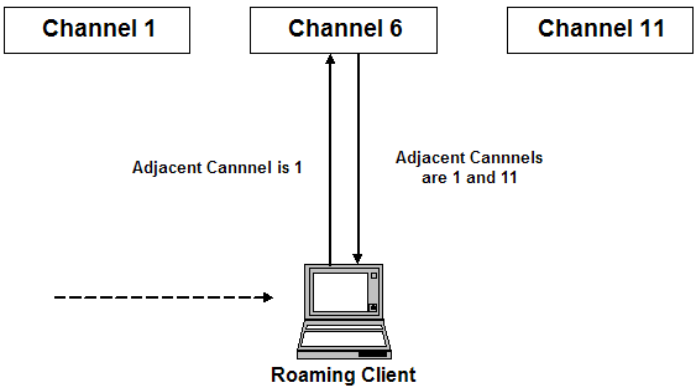
2) Ha a potenciális cél AP repeater módban van, és több rádió hop-ra van a gerinchálózattól, mint az aktuális AP, akkor 20%-kal nagyobb jelerőssége kell, hogy legyen, mint a jelenlegi AP-nak.

3) A potenciális cél AP-nál a küldő egység terheltsége maximum 10%-kal lehet nagyobb, mint a jelenlegi AP esetén.

A terminál a felsorolt alapkritériumoknak megfelelő bázisállomásokat összehasonlítja a jelenlegi bázisállomással. Ha egy elfogadott AP teljesít egyet az alábbi feltételek közül, akkor azt a terminál új, aktuális AP-nak választja,

majd a lista többi AP-ját már ehhez az újonnan választott AP-hoz hasonlítja a továbbiakban: a jelerősség 20%-kal nagyobb, mint az aktuális bázisállomásé; kevesebb hop távolság a gerinchez; legalább néggyel kevesebb a kapcsolódott kliensek száma, mint a jelenlegi AP esetén; legalább 20%- kal kisebb a küldő egység terheltsége.

A 12.2.(11)JA IOS verziótól kezdődően a Cisco „fast secure roaming" implementáció két újabb lehetőséggel bővült: egyrészt növelt hatékonyságú a 802.11-es csatornakeresés a fizikai roaming alatt, másrészt hatékonyabb újra hitelesítési mechanizmus jelenik meg, amely fejlett titkosító kulcs menedzsmentet alkalmaz[24]. Függetlenül az alkalmazott biztonsági módszertől, a hatékonyabb csatornakeresés gyorsabb L2 roaming-ot tesz lehetővé.



21. ábra A fast roaming csatorna keresése

Az újrahitelesítés hatékonyságát növelő kulcs menedzsment felgyorsítja a Cisco LEAP hitelesítési folyamatot, így a roaming rövid idő alatt és biztonságosan zajlik le. A Cisco terminálokon és bázisállomásokon az IEEE 802.11 csatornakeresés alapértelmezés szerint egyaránt engedélyezett. A "fast secure roaming"-ot egy csatornakeresés előzi meg. A 12.2(11)JA előtti IOS verziók esetén a kliensnek 37ms vett igénybe egy rádiócsatorna ellenőrzése, ami a magyar szabványok szerinti 13 csatorna esetén összességében 481ms-ot jelent. A kliens minden egyes csatorna esetén az alábbi lépéseket hajtja végre: miután a terminál rádiós hardvere ráhangolódik az adott WLAN csatornára, figyel hogy elkerülje az ütközést, majd „probe" keretet küld és várja a „probe response" vagy a „beacon" jelzést.

A fast secure roaming esetén hatékonyabb a csatornakerés: az újrahitelesített kliens informálja az új AP-t a korábbi AP-val való kapcsolat elvesztése óta eltelt időről, a csatornaszámról, és az SSID-ről. Ezeket az információkat felhasználva, az új AP felépít egy listát a szomszédos bázisállomásokról, és az általuk használt rádiócsatornákról.

Ha a szomszédos AP-król információt szolgáltató mobil terminál több, mint 10 másodperce kapcsolódott le az előző AP-ról, akkor az általa küldött információkat nem veszi figyelembe az új AP. A bázisállomások maximum 30 szomszédos AP-ról tárolnak információt. Ez a lista egy egynapos periódus alatt elévül. Amikor a terminál asszociál egy AP-hoz, az új bázisállomás unicast csomagban visszaküldi számára a szomszédos AP-k listáját. Ha a kliensnek roaming-ot kell végrehajtania, megvizsgálja az aktuális AP-tól kapott listát, és csak azokat a rádiócsatornákat ellenőrzi, melyeket a szomszédos bázisállomások valamelyike használ.
A kliensállomás az elfoglaltságától függően az alábbi három roaming típus egyikét alkalmazza:

- *Normal roam*: a kliens nem kapott és nem küldött unicast csomagot az elmúlt 500 ms-ban. Nem használja az AP-tól kapott listát, ellenőrzi az adott térségben érvényes összes 802.11-es csatornát.

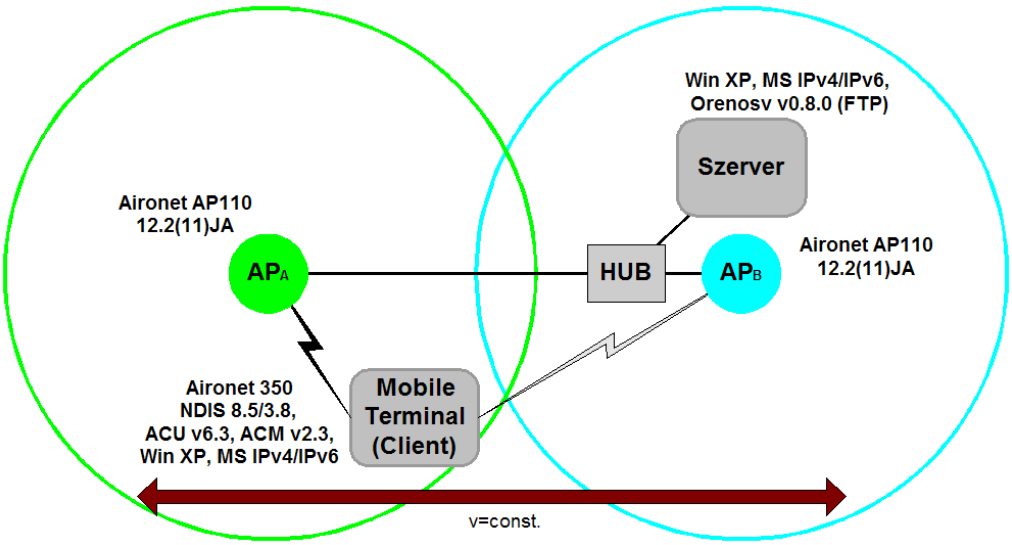
- *Fast roam*: a kliens kapott vagy küldött unicast csomagot az elmúlt 500 ms-ban. A szomszédos AP-k által használt csatornákat ellenőrzi. Ha nem talál új AP-t a lista alapján, akkor átvizsgálja az összes csatornát. A kliens 75 ms-ra korlátozza a keresési idejét, ha legalább egy jobb AP-t tudott találni.

- *Very fast roam*: a kliens kapott vagy küldött unicast csomagot az elmúlt 500 ms-ban, és nullánál nagyobb százalékkal növeli az adott cella terheltségét. A többi esemény a „fast roaming”-gal megegyező kivéve, hogy jobb bázisállomás találata esetén a keresés azonnal befejeződik.

A méréseinkhez olyan eszközparkot használtunk, amely a fenti három roaming típus bármelyikét végre tudja hajtani.

7.4. Mérési környezet

Az IPv4 és IPv6 protokollok viselkedését mobil környezetben úgy vizsgáltuk, hogy egy kültéri teszt WiFi rendszert állítottunk össze. Ez IEEE 802.11b szabvány szerint működő két darab egymástól 100 méteres távolságon huzalos Ethernet kapcsolattal összekötött bázisállomásokból és egy mobil terminálból (kliensből) állt. Mint ismeretes a 11 Mbps-os WiFi szabvány is támogatja a roaming funkciót. Ugyanakkor a vezetéknélküli adatátvitel sebessége erőteljesen függ a bázisállomás és a kliens közötti távolságtól. A mobil WiFi terminál mozgás közben közeledik, majd távolodik a bázisállomástól.



22. ábra A mérési környezet

Ez az adatkapcsolati rétegben az átviteli sebesség automatikus váltását okozza a 0:1:2:5,5:11 Mbps-os értékek között. Alapértelmezés szerint roaming esetén 11:5,5:2:1:0:1:2:5,5:11 Mbps-os sebességértékek mellett történik az átvitel. A mi esetünkben az átviteli sebességet 11 Mbps-ra rögzítettük, ezáltal a térerő változása kényszerítette a terminált a roaming kezdeményezésére.

A szállítási réteg különböző protokolljainak viselkedését figyeltük, miközben a mobil terminál egy autóban - a roaming idején konstans sebességgel - mozgott

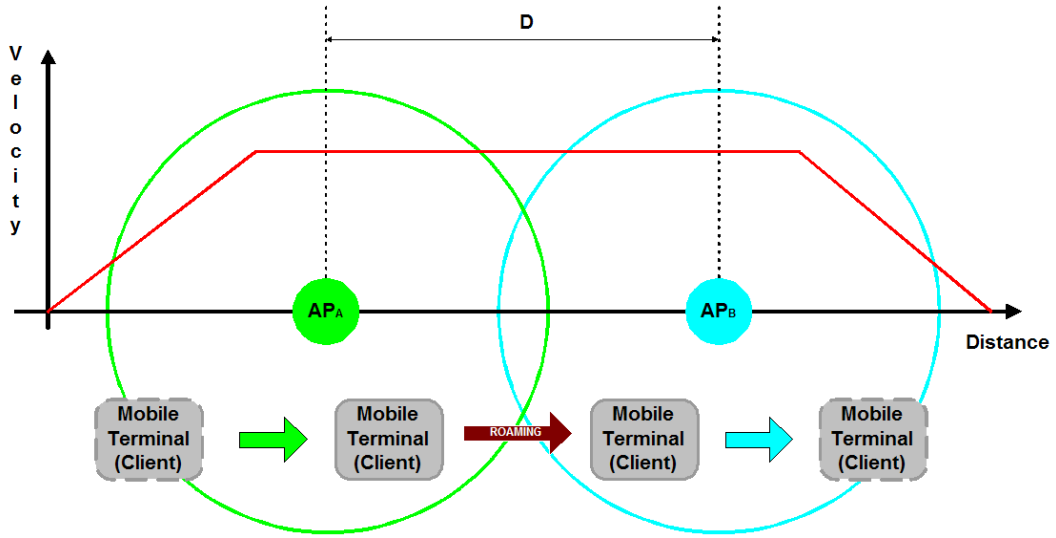
a bázisállomások közötti iránnyal párhuzamos útvonalon. A szerver oldalon az Ethereal snoop programot futtattuk, amely az adatkapcsolati réteg minden egyes keretét időbélyeggel letárolta és további analizálásra adott lehetőséget.

Roaming idején az adaforgalom iránya fontos, hiszen ebben az időintervallumban különböző módon viselkedik a TCP az adatfolyam irányának függvényében:

- **Forgalom a vezetékes oldalról a vezeték nélküli kliens irányába:** A vezetékes oldal nem tud a roaming eseményről, nincs információ a kiesésről, így nem szünetelteti a csomagok küldését. Ennek a következménye, hogy a kliens cellaváltása után újra kell küldenie az elveszett csomagokat.

- **Forgalom a vezeték nélküli kliens irányából vezetékes irányba:** A vezeték nélküli kliens saját protokoll stackjén keresztül információval rendelkezik az alatta lévő rétegekben végbemenő roaming eseményről, így a kiesés idejére felfüggeszti a csomagok küldését, így nem alakul ki csomagvesztés, nincs újraküldés.

Az ICMP üzenetek méretét úgy választottuk meg, hogy a spay ping esetén 64 bájt, 1500 bájt, illetve 32 Kbájt méretű legyen az adatkapcsolati keret mérete. Ennek jelentősége a minimális, maximális keretméret (MTU), illetve az IP csomag szegmentálásánál van. Az autó sebessége a két bázisállomás közötti szakaszon konstans volt és a lakott területen szokásos értékek szerint közlekedtünk (10 Km/h, 30 Km/h, 50 Km/h).



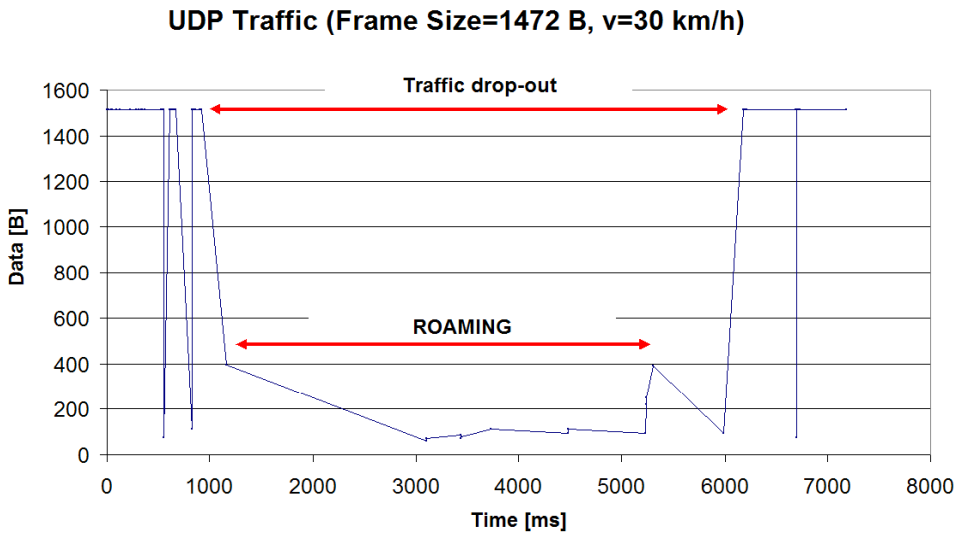
23. ábra. A roaming folyamat

Paraméterek	
Access Point	Cisco Aironet AP1120
Mobile Terminal	Cisco Aironet 350 series
AP IOS (1 mW)	12.2(11)JA
MT and server OS	Windows XP
MT Radio Firmware	Win/NDIS Driver 8.5/3.8, ACU v6.3, ACM v2.3
FTP server (IPv4/IPv6)	Orenosv v0.8.0

Független mérések	
L4 protocol	TCP (FTP), UDP (Spray)
L3 protocol	IPv4, IPv6
TCP traffic	MT->Server (Up) Server->MT (Down)
UDP message [B]	18, 1472, 31970
Speed [Km/h]	10, 30, 50
D(APA,APB)	100 m

7.5. Új eredmények

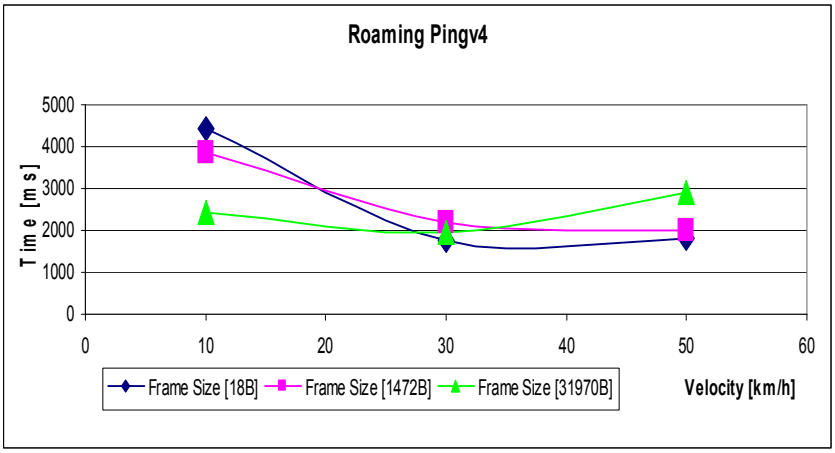
A capture programmal vételezett keretsorozatból értelmezni lehet a roaming folyamat jelzését, valamint a szállítási réteg forgalmát. Ezáltal mérhetővé vált a roaming R[ms], illetve az ebből származó forgalom kiesés T[ms] időtartama[J1].



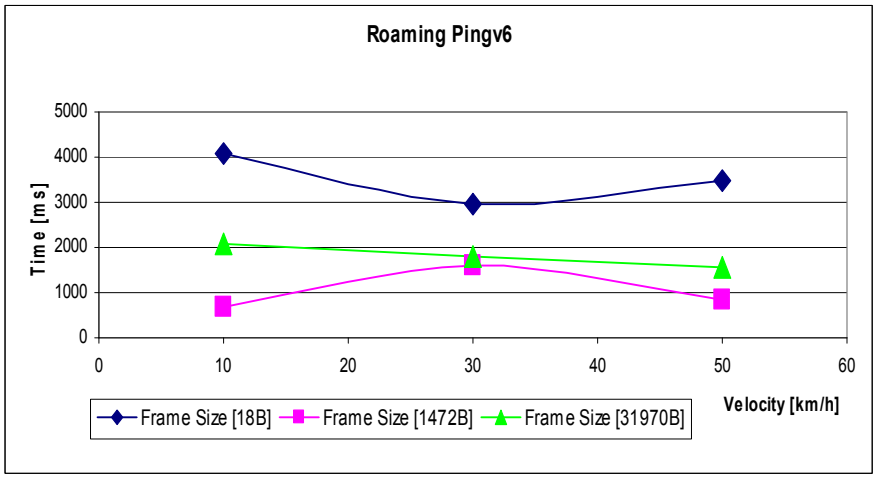
24. ábra A mért paraméterek (R[ms], T[ms])

A TCP kapcsolatok „Slow Start” és „Windowing” algoritmusa alapján valósul meg a nagyméretű fájlok továbbítása FTP-vel. Az adatkapcsolati réteg átviteli sebességének változása a window méretének szabályozását teszi szükségessé. A WiFi átviteltechnika roaming fázisának időtartama erőteljesen befolyásolja a TCP hatásfokát. Az UDP átvitel a jellegéből adódóan sokkal alkalmazkodóbb természetű. Másodpercenként 100 csomagot küldtünk a spay ping segítségével, amely a csomagmérettől (64 bájt, 1500 bájt, 32 kbájt) függően a rádiós csatornát 0,93%, 21,82%, illetve 100%-ig tehelte. A mért paraméterek alapján a következő megállapításokat tehetjük:

Alapértelmezett MTU alatti Ethernet keretméret (1500 bájt) esetén az ICMPv4 roaming ideje csökken a sebességgel, míg MTU feletti keretméret esetén ugyanez növekvő tendenciát mutat. Szegmentációnál több időbe telik a csomagok sorrendjének visszaállítása. Az ICMPv6 roaming ideje minden keretméret esetén gyakorlatilag csökken a sebességgel. Ez az IPv6 közegérzékelő tulajdonságának tulajdonítható.



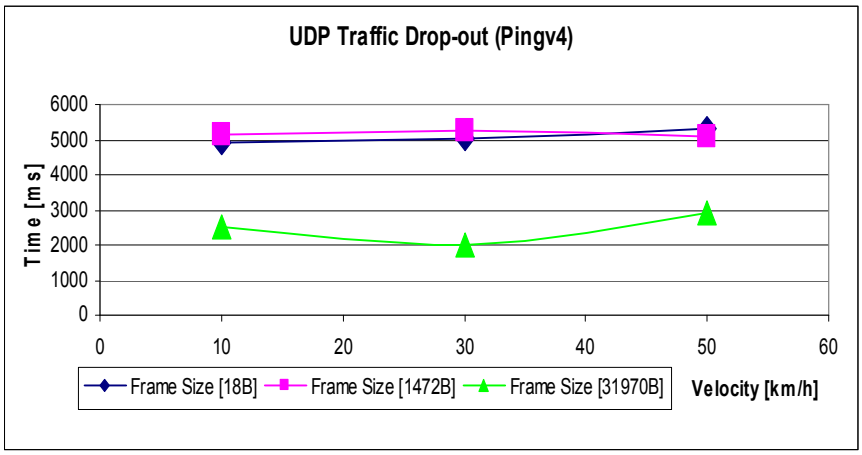
25. ábra Effect of the frame size to the roaming (ICMPv4)



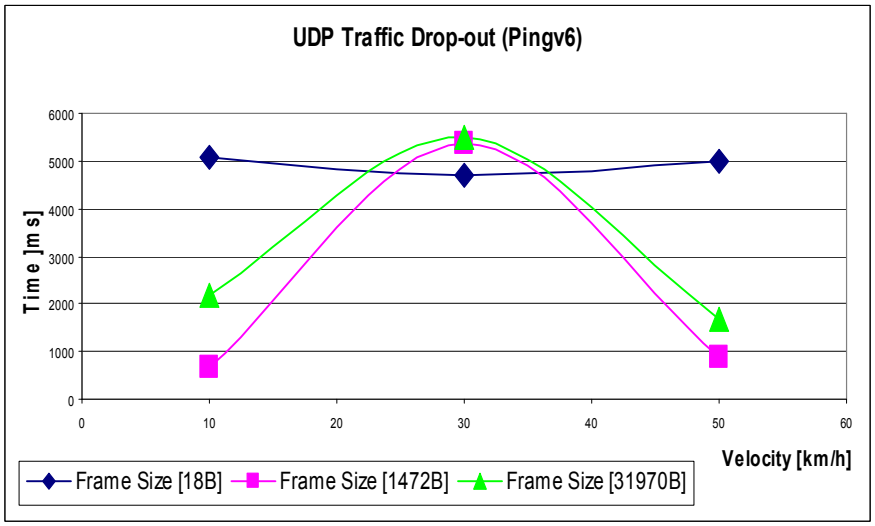
26. ábra Effect of the frame size to the roaming (ICMPv6)

Ethernet alapértelmezett MTU keretméret alatt az ICMPv4 forgalom kimaradása a sebességtől független, de a szegmentáció az időkésleltetést csökkenti. A látszólag ellentmondásos hatás a csatorna folyamatos terhelésével magyarázható. Az ICMPv6 esetén erőteljes ingadozás tapasztalható a sebesség függvényében. Ennek oka az IPv6-nak az L2 rétegtől való lekapcsolódása miatt

adódik. Ezt a lekapcsolódást az IPv4 nem teszi meg, így kevésbé érzékeny az ICMPv4.

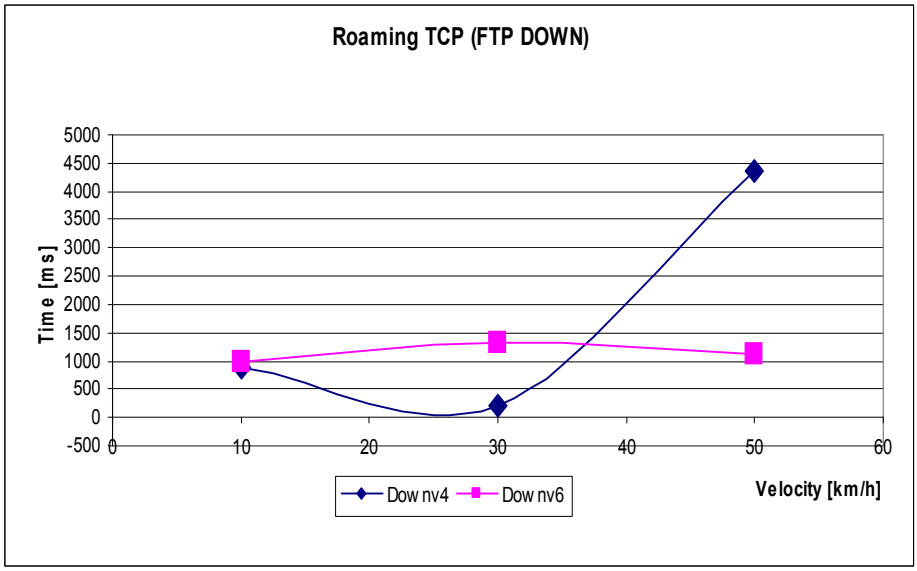


27. ábra Effect of the roaming to the UDP v4

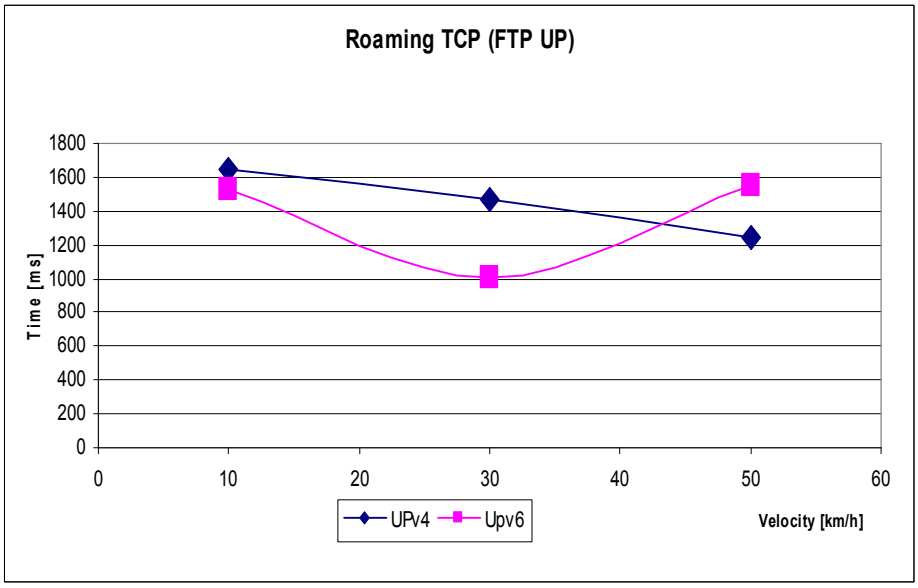


28. ábra Effect of the roaming to the UPD v6

A forgalom kimaradás ICMPv4 esetén másfél másodperccel, ICMPv6 eetén pedig csak egy másodperccel hosszabb, mint a roaming időtartama. Kis keretméret esetén kevésbé függ a sebességtől az ICMP forgalom kimaradása, míg nagyobb keretméretek esetén csak az ICMPv6 érzékeny a sebességre.



29. ábra Effect of the roaming (TCP download traffic)

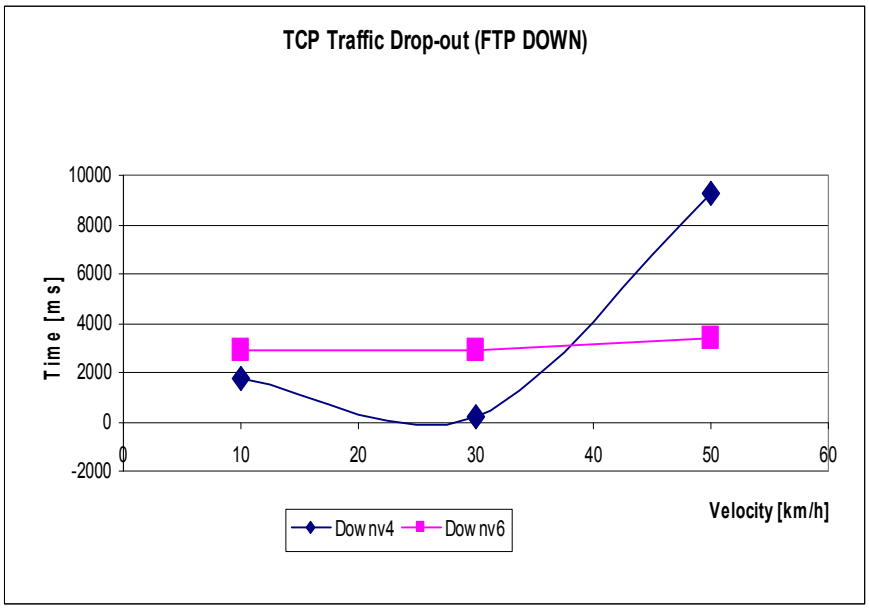


30. ábra Effect of the roaming(TCP upload traffic)

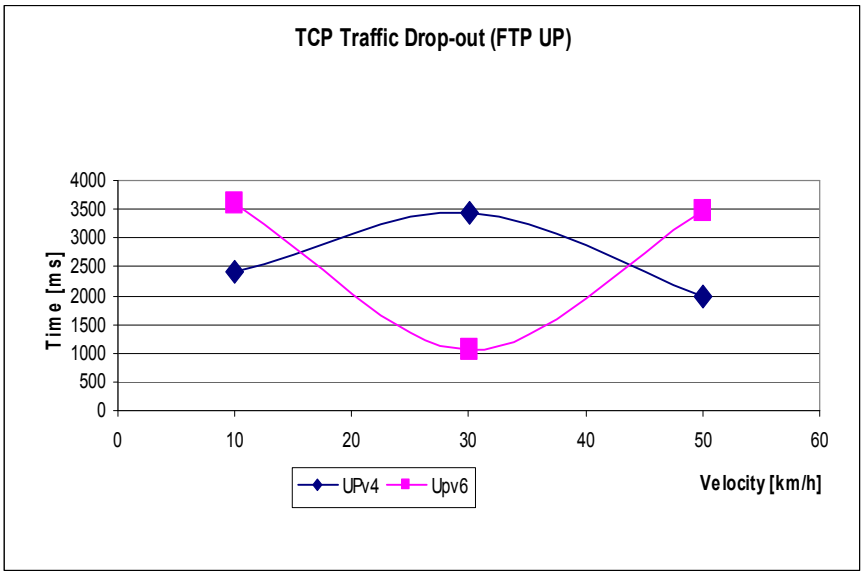
A forgalom kimaradás TCPv6 esetén független az adatfolyam irányától. TCPv4 esetén azonban a letöltés szignifikánsan nagyobb adatkiesést szenved, mint a feltöltés esetén. Ez abból adódik, hogy a TCPv4 sokkal gyorsabban

változtatja a window méretét, így letöltéskor sok adat elvész a régi bázisállomás irányába küldött jelentős mennyiségű keretszám miatt.

A TCPv4 nagyobb ablakmérettel és kisebb dinamikával dolgozik, míg a TCPv6 kisebb ablakméretket alkalmaz és gyorsan szabályozza azt. Emiatt a TCPv6 jobban viseli a WiFi környezet roaming eseményeit, csökkentve ezáltal a forgalom kieséseket.



31. ábra Effect of the roaming (TCP download traffic)



32. ábra Effect of the roaming (TCP upload traffic)

A TCPv4 forgalom kimaradása lefelé irányú adatforgalom esetén erőteljesen függ a mobil terminál mozgási sebességétől. Ötven kilométeres óránkénti sebességnél akár 9,2 másodperces kiesést is képes produkálni. Ez lehetetlenné teszi a gyors járművekből történő folyamatos kommunikációt. TCPv6 lefelé forgalomnál ez az érték gyakorlatilag független a mozgási sebességtől és 3,8 másodperc alatt marad. A TCPv4 forgalom kimaradása felfelé irányú adatforgalomnál kis mértékben módosul a terminál sebességével, míg ugyanez TCPv6 esetén jelentősen változik.

7.6. Új eredmények értelmezése

A mobil kliens bázisálomásokhoz viszonyított relatív sebessége és a roaming végrehajtásának kölcsönhatása jelentősen befolyásolja a TCP kapcsolatokat, miközben kevésbé hat az UDP átvitelre. Az összehasonlító mérésekből statisztikai módszerekkel nyert eredmények lehetővé teszik, hogy valós képet kapjunk az IPv4 és az IPv6 mobil átvitel esetén tanusított viselkedésére vonatkozóan, valamint választ kaphatunk arra a kérdésre, hogy valóban magasabb minőségű mobil adatátvitelt eredményez-e az IPv6 protokoll vezeték nélküli adatkapcsolati réteg fölött elődjéhez, az IPv4-hez képest. A hagyományos elektronikus alkalmazások az IPv4 protokoll „best effort” jellege

miatt lasúbb átvitelt biztosítanak mobil WiFi környezetben, míg az IPv6 protokoll az alsóbb rétegekhez történő gyors adaptáció miatt hatékony átvitelt képes biztosítani.

Az időérzékeny alkalmazások (IP telefon, videokonferencia, stb.) az IPv4 protokoll QoS korlátai miatt mobil WiFi környezetben nagy kieséseket szenvednek, így a minőség elfogadhatatlan. Az IPv6 gyors adaptációja miatt a kiesések kisebbek, ezért a jelenlegi mobil WiFi környezetben fast roaming esetén közel elfogadható minőségű infokommunikációs szolgáltatások használhatók[J1].

A témával kapcsolatosan további elemzési lehetőség a lakott területen kívüli környezetben, nagyobb mozgási sebességgel haladó mobil terminálok (autópályán, vonaton) adatkommunikációs szolgáltatásainak minőségét befolyásoló tényezők feltárása és értelmezése. Egyértelműen körvonalazódik, hogy a vezeték nélküli kapcsolatok mobil funkcióinak kiaknázásához halaszthatatlan egyrészt a roaming folyamat gyorsítása, másrészt pedig az IPv6 feletti speciális alkalmazások kifejlesztése.

8. TCP alapú multimédia alkalmazások vizsgálata WiFi hálózatokon

Az IEEE 802.11 családhoz tartozó vezeték nélküli adatátviteli mechanizmusok a mobilitás miatt széles körben terjedtek el úgy beltéri, mint kültéri környezetben. A hot-spot-ok kialakításánál alapvető kérdésként vetődik fel, hogy a 802.11b/g és/vagy a 802.11a szabványnak megfelelő rendszer telepítésére kerüljön sor. Ennek eldöntése gazdasági racionalitási megfontolásokon túlmenően hatékonyság elemzést is szükségessé tesz[25].

Mint ismeretes, a WiFi rendszer az ISM frekvencia sávokra épül, ami lehetővé teszi, hogy ugyanazon fizikai környezetben egymástól függetlenül akár több szolgáltató is hotspot-okat telepítsen. A gyakorlati tapasztalat szerint kültéri környezetben a különböző szolgáltatók a használt rádiós csatornákat egymás között egyeztetés nélkül, vagy csak ritkán egyeztetett formában használják. Mivel a kisugárzott mikrohullámú energiára ETSI szabványok vonatkoznak, a sűrűn telepített WiFi rendszerek egymásra zavaró hatással vannak. Céges, illetve egyetemi környezetben egyre hangsúlyosabban fogalmazódik meg az igény, hogy a WiFi mobil eszközök (notebook, palmtop, intelligens mobil telefon) multimédiás szolgáltatásokat is biztosítsanak. Mivel egyetemi környezetben egyre jobban elterjednek az IP telefon rendszerek, egyértelmű feladatként jelenik meg a WiFi telefonok campus területén beltéri, illetve kültéri környezetben, mozgás közbeni használhatóságának elemzése. A 2,4 GHz-es ISM tartományban a WiFi IP telefon beszédtovábbítási tulajdonságai a hangkódolási algoritmustól függnek. Az 5 GHz-es WiFi átvitel speciális csatornakódolási mechanizmusa hatékonyabb, mint az IEEE 802.11g esetén, ugyanakkor az átviteli sebesség nagyon érzékeny a bázisállomástól mért távolságra.
Mozgás közben a nagyobb tömörítési aránnyal működő adatátviteli szabvány érzékenyebb a rádiós cellák közötti váltásra, mint az alacsonyabb tömörítésű algoritmus.

Előzetes elemzések alapján ismerjük, hogy a mobil terminálokon használható multimédiás szolgáltatások minőségét erőteljesen befolyásolja a készülék roaming közbeni fizikai mozgásának sebessége [C2].
A mobil terminálokon működő multimédiás alkalmazások minősége erőteljesen függ az adatkapcsolati rétegben lejátszódó folyamatoktól.

8.1. Multimédia codec technológiák áttekintése

A DSP (Digital Signal Processing) architektúrák utóbbi években bekövetkezett látványos fejlődése, valamint a humán beszédfelismerés területén végzett kutatásoknak köszönhetően a hangkódoló/dekódoló (codec) technológiák komoly előrelépést tettek. Az új kodekek az egyszerű AD/DA átalakításon túlmenően, a becslő minták alkalmazása segítségével a bemenő hangjelet analizálják és minimális sávszélességet igénylő
adatfolyamként képesek tovább küldeni.

8.1.1. PCM

Az egyszerű PCM (Pulse Code Modulation) kódolású hang az ITU-T G.711-es szabvány szerint történik. A 64 kbps-os PCM hang tömörítése a μ-law és az A-law eljárásokkal történik úgy, hogy a 12, 13 bites mintavételt logaritmikus törvény szerint képezi le 8 bitre. A két leképezési törvény analitikus formája az alábbiak szerinti.

Előnyök: egyszerű, kis komplexitású, kis késleltetés, jó hangminőség. Hátrány: nagy sávszélesség igény.

$$y = \mathrm{sgn}(x) \cdot \frac{\ln(1+\mu \cdot |x|)}{\ln(1+\mu)},$$

ahol:
y – normalizált kimenet, [-1, 1] között
x – normalizált mintavétel, [-1, 1] között
μ = 255, kompressziós paraméter

$$y = \begin{cases} \dfrac{A}{1+\ln(A)} x, & ha \quad |x| \le \dfrac{1}{A} \\ \dfrac{\mathrm{sgn}(x)}{1+\ln(A)} \cdot (1+\ln|Ax|), & ha \quad \dfrac{1}{A} < |x| \le 1 \end{cases},$$

ahol:
y – normalizált kimenet, [-1, 1] között
x – normalizált mintavétel, [-1, 1] között
A = 255, kompressziós paraméter

8.1.2. ADPCM

Az ADCPM (Adaptive Differential Pulse Code Modulation) ugyancsak gyakori kompressziós megoldás, amely az ITU-T G.726 szabványban van rögzítve. Ez négybites mintákat alkalmaz, amelyeket 32 kbps-os szállítási sebességgel továbbít. A PCM-mel ellentétben a négybites szók nem közvetlenül a beszéd amplitúdóját kódolják, hanem az amplitúdók különbségét és a változások rátáját.

Ehhez egy nagyon egyszerű lineáris becslést alkalmaz. Előnyök: egyszerű, kis komplexitású, jó minőségű hang, kis késleltetés, több kódolási sebesség. Hátrányok: viszonylag nagy sávszélesség igény, kis sávszélességen a hang minősége romlik.

8.1.3. AMR-NB

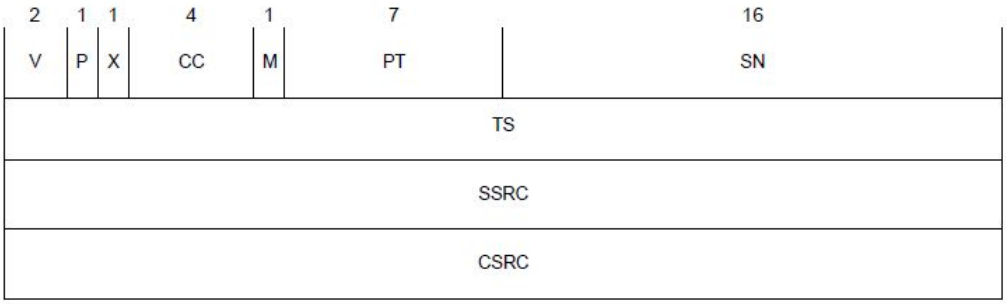
Az AMR (Adaptive Multi Rate - Narrow Band) a GSM és UMTS mobilhálózatokban használnak. Az algoritmus nyolc kompressziós arányt támogat (4,75; 5,15; 5,90; 6,70; 7,40; 7,95; 10,20; 12,20 kbps). Az algoritmus bármikor képes váltani ezen arányok között, ami IP alapú hálózatokban előnyt jelent. A küldő bármikor megváltoztathatja a kimenő sávszélességet az RTP által valós időben szolgáltatott statisztikák alapján: az RTP réteg visszajelzésére a kódoló a következő hangmintákat már a megváltozott kódolási sebességgel tömöríti, és ezt a dekóder ugyanúgy dekódolni tudja. A 20ms-os műsorkeretek kódolása ACELP algoritmus alkalmazásával, és 5 ms lookahead értékkel történik. Előnyök: egyszerű, viszonylag kis komplexitású, kis sávszélesség igény, kis késleltetés, jó hangminőség, több kódolási sebesség. Hátrányok: még kevés implementáció létezik, nincs nyílt forráskód.

8.1.4. AMR-WB

Az AMR-WB (Adaptive Multi Rate - Wide Band) mechanizmust a G.722.2 kódoló alkalmazza, amely nagy sávszélességre optimalizált ACELP algoritmust használ és 7 kHz-es hangjelet kódol 16 kHz-en mintavételezve. Adaptívan változtatja a kódolási sebességet (23,85; 23,05; 19,85; 18,25; 15,85; 14,25; 12,65; 8,85; 6,6 kbps). A kódoló 20ms hosszúságú kereteket használ és 5ms lookahead buffert használ. Előnyök: nagyon jó minőségű hang, kis késleltetés, több kódolási sebesség. Hátrányok: nagy sávszélesség igény a jó minőségű hanghoz, közepesen nagy számolási komplexitás.

8.1.5. Az RTP protokoll

Az RTP (Real-Time Protocol) valós idejű forgalom számára végponttól-végpontig terjedő szállítási szolgáltatást (hang, kép) biztosít. Ehhez olyan szolgálatokat vesz igénybe, mint a PDU azonosítás, sorszámozás, időbélyegzés, és az átvitel felügyelete. Az RTP protokoll alkalmazás szintű keretezést valósít meg. Általában UDP felett alkalmazzák, felhasználva annak multiplexelési és ellenőrző összeg képzési szolgáltatásait, de ritkán TCP felett is működtetik. Az RTP nem garantálja a csomagok megérkezését, és a helyes sorrendben érkezést sem. A mechanizmusnak két része van, az RTP és az RTCP (Real-Time Control Protocol). Az RTP PDU-k a valós idejű adatot szállítják, míg az RTCP PDU-k az átvitel minőségére és az entitásokra vonatkozó vezérlő
információkat továbbítanak. Az RTP az IP hálózatokra jellemző változékony és túlterhelt hálózati feltételre van optimalizálva. Az RTP a tartalom adatokat továbbítja egyik irányba és az RTCP kétirányú csatornáit használja a minőségi jellemzőket is magába foglaló vezérlő információk számára. Az RTP viszony kiépítésekor az alkalmazások meghatározzák úgy az RTP, mint az RTCP számára a műsor csatornánkénti a szállítási címet. Ez entitásonként az IP hálózati cím és a portszám páros lesz. Minden RTP csomagnak fix szerkezetű fejléce van, amelyet a 33. ábra szemléltet.



33. ábra RTP fejléc

Az első tizenkét bájt minden RTP csomagban megtalálható, viszont a közreműködő forrás azonosítók listája (CSRC) csak akkor fordul elő, ha azokat a keverő elhelyezte a csomagba. A mezők jelentése az alábbi:

- *V*: verzió (jelenleg 2)
- *P*: kitöltés (Padding), ha a bit értéke 1, akkor a csomag végén vannak kitöltő bájtok, amelyek nem a tartalom

adat részei. Az utolsó kitöltő bájt tartalmazza, hogy hány kitöltő bájtot kell figyelmen kívül hagyni.
Kitöltésre lehet szükség, például fix blokkméretű titkosító algoritmus alkalmazásánál.
- *X*: kiterjesztés (Extension) : ha értéke 1, akkor a fix fejléc után következik pontosan egy fejléc kiterjesztés.
- *CC*: közreműködő forrásszámláló (CSRC Count): a fix fejléc után következő közreműködő forrás azonosítók száma.
- *M*: jelző (Marker) : a jelző bit értelmezése az alkalmazás profilban van meghatározva. Jelezheti például a
képkockák határát a csomagfolyamban.
- *PT:* tartalom adat típus (Payload Type) : az alkalmazás profilban adott, hogy a típuskódhoz milyen tartalom
adat formátum tartozik. Egy RTP adó egy adott tartalom adat típust bocsát ki egy viszonyban.
- *SN:* sorszám (Sequence Number) egyesével növekszik, minden elküldött csomaggal. A vevő ezáltal tudja
észlelni, ha csomagvesztés történt, illetve helyre tudja állítani a sorrendet. Biztonsági okokból a kezdeti
értéke véletlengenerált szám.
- *TS:* időbélyeg (Time Stamp) az RTP csomag adatrészében található első bájt mintavételezési időbélyege
monoton és lineárisan növekvő órától származik. A kezdőérték itt is véletlengenerált szám. Egymás utáni RTP csomagoknak lehet ugyanaz az időbélyege, ha egyszerre keletkeztek, például ha ugyanahhoz a képkockához tartoznak. Az egymás után küldött csomagokban található időbélyegek nem feltétlenül monoton növekvők, ha az adatok nem a mintavételezésük sorrendjében kerülnek továbbítására, mint például az MPEG interpolált képkockáinál.
- *Szinkronizációs forrás* (SSRC) azonosító: azonosítja a forrást szinkronizáció céljából. Véletlen módon választott azonosító, minden forrásra egyedi. Ha megváltozik a szállítási cím, akkor meg kell változtatni az SSRC azonosítót is.
- *Közreműködő forrás* (CSRC) azonosító: 0-15 db, egyenként 32 bit, azonosítja az adatfolyamhoz tartozó közreműködő forrásokat. Ezt a keverő helyezi el a fejlécben, a közreműködő források SSRC azonosítóit felsorolva, így a vevő azonosítani tudja az adókat.

E mezők felhasználásával az RTP olyan funkciókat tud ellátni, mint az idő helyreállítás (időbélyeg mező), adóazonosítás (SSRC), tartalom azonosítás (PT), sorszámozás, veszteség észlelés. Nem az RTP hatáskörébe tartoznak a szolgáltatás minőség garantálása, az erőforrás foglalás, az időben történő kézbesítés, valamint a csomagvesztés helyrehozása. Mindezek mellett az RTP alkalmas valós idejű tartalom szállítására.
Az RTCP-t az RTP-vel együtt használják és elsősorban az RTP átvitelének a monitorozására, illetve szabályozására szolgál. Célja az adatátvitel minőségéről és a viszony résztvevőiről való értesítés. Az RTCP működése a szabályozó csomagok viszonybéli összes résztvevőnek való időnkénti újraküldésén alapul. Az RTCP is UDP felett fut. Több fajta RTCP csomag van, amelyek a vevő jelentést, az adó jelentést, a forrás leírást, a kapcsolatot bontást és az alkalmazásra jellemző feladatkör információkat tartalmazza. A különböző típusú csomagok szerkezete eltérő, viszont több különböző csomagot egybe lehet fogni, és együttesen lehet elküldeni.

8.1.6. Hang kódolók/dekódolók csoportosítása

A PCM és az ADPCM a hullámforma kodekek csoportjába tartoznak, amelyek a hullámforma redundáns karakterisztikáit használják fel. Az utóbbi 10-15 évben kifejlesztett más kompressziós technikák a beszéd forrás karakterisztikáira építenek. Ezek jelfeldolgozás és tömörítés segítségével az eredeti beszédjelnek csak az egyszerűsített paramétereit küldik el, így kisebb sávszélességet igényelnek. Ezeket forrás kodekeknek nevezzük és ide tartoznak az LPC (Linear Predictive Coding), a CELP (Code Excited Linear Prediction), valamint az MP-MLQ (Multipurpose Multilevel Quantization) eljárások. A fejlett becslő kodekek az emberi beszédjel forrást matematikai modellel helyettesítik és tömörített hangküldés helyett a hang reprezentációját továbbítják. A legnépszerűbb telefon hangkódolási és csomagkapcsolt hang szabványok az alábbiak:

- *G.711:* A 64 kbps PCM hangkódolási technika, amely a hagyományos digitális PBX központokban, illetve hálózatokban használatos.
- *G.726:* Ez 40, 32, 24, 16 kbps-os ADPCM kódolást használ. Az ADPCM hang a csomagkapcsolt és a hagyományos PBX hálózatok közötti hangátvitelhez javasolt.

- *G.728:* Ez a CELP tömörítés kis késleltetés ingadozásos változatával 16 kbps-os sávszélességen továbbítja a beszédet. A CELP hangot transzkódolni kell

nyilvános telefon formátumra ahhoz, hogy nyilvános végpontokkal sikeres kommunikációs jöhessen létre.

- *G.729:* Ez CELP tömörítéssel a hangot 8 kbps-os jelfolyammá alakítja. A két alváltozata a processzálás komplexitásában lényegesen különbözik egymástól, és mindkettő a 32 kbps-os ADPCM-nek megfelelő beszédminőséget biztosítja.

- *G.731:* Ez beszéd vagy multimédiás szolgáltatás hang komponensének tömörítését végzi, nagyon alacsony sávszélesség mellett. A H.324 protokoll család részeként az 5,3 kbps, illetve a 6,3 kbps sávszélességen dolgozik. Előbbi CELP, utóbbi pedig MP-MLQ technológiát alkalmaz, miközben jó minőségű beszédátvitelt és további rugalmasságot biztosít a rendszer számára.

- *GSM:* A GSM (Global System for Mobile Communications) az ETSI I-30036 szabványa és széles körben használt, az európai mobil rádióhálózatokban hang és kis sávszélességű adatkommunikációra. A GSM teljes sebességű hangkódoló 13 kbps sebességen működik és RPE (Regular Pulse Excited) kódolót használ 8 kHz mintavételezési frekvencia mellett. A félsebességű GSM kódoló 7 kbps sávszélességet igényel 5 kHz mintavételezés mellett. A bemeneti hang 20 ms hosszúságú keretekre van osztva és minden keretre 8 rövid 4 távú becslést végeznek. Ezután minden keret további 5 ms hosszúságú alkeretekre bomlik, melyekre a kódoló késleltetést és nyereséget számol a hosszú távú becslő számára. Végül a maradék jelet kvantálja minden alkeretben. A GSM kódoló jó minőségű hangot generál, mindazonáltal a G.728 kódoló (CELP) mégis felülmúlja a nagyobb sávszélességgel. A GSM kódoló kis számításigényű. Előnyök: egyszerű, viszonylag kis komplexitású, kis sávszélesség igény, kis késleltetés, nyílt forrás. Hátrányok: a sávszélesség/hangminőség arányban a G.729 felülmúlja.

8.1.7. A Nullsoft Video protokoll

A Nullsoft Video (NSV) formátum egy olyan bitstream jelfolyam, amely képes biztosítani a hang és videó közös becsomagolását. A gyakorlatban alkalmazott mindegyik hang és videó tömörítési mechanizmussal együttműködik. Mivel bitstream formátum, így nem igényli a teljes fájl letöltését a lejátszáshoz. Képes streaming szolgáltatásra, megbízható szinkronizálás valósul meg a jelfolyam bármely pontján. Másodlagos adatcsatornák segítségével több hang, feliratozás, vagy más adatfolyam is biztosítható. Az NSV fájl szerkezete két fő részből áll: egy opcionális fájl fejrész és egy kötelező

bitstream alkotja. Minden több bájtos egész szám LSB formátumban van ábrázolva, azaz a legkisebb helyiértékű bájt baloldalon helyezkedik el. Így egy négy, illetve egy húszbites szám három bájtot fog elfoglalni.

Az NSV fájl fejrész formátuma: Az NSV fájlnak csak egy fájl fejrésze lehet, amely tartalmazza a fájl méretét bájtokban és ezredmásodpercben, a tartalomjegyzéket, amely a VBR tartalom szabatos keresését biztosítja, és a metaadatokat (34. ábra).

34. ábra NSV formátum fejrésze

A fájl fejrész tartalmazhat további olyan információkat, mint a műsor címe, szerzője, javasolt képernyő oldalméreteinek aránya, stb. A metaadat bármennyi név-érték párt tartalmazhat. *Az NSV fájl fejrész tartalom tábla (TOC- Table of Contents) formátuma:* Négybájtos egész számok tömbje. A TOC v1.0 esetén a bejegyzés sorszáma a bejegyzés idejével arányos. A bejegyzés értéke képezi az NSV bitstream-ben elfoglalt offset pozíciót. Nagyobb fájl esetén a keresés pontatlan volt. A TOC v2.0 esetén viszont adott bejegyzés a kulcskeret offset-jét adja a bitstream részletben, míg a tartalom méretével növelt sorszámú bejegyzés a kulcskeret abszolút helyét mutatja. Ez pontos keresést tesz lehetővé.

Az NSV bitstream formátum: A jelfolyam NSV kereteket tartalmaz, amelyek lehetnek szinkronizációs vagy nem-szinkronizációs keretek. Az NSV jelfolyam legalább egy szinkronizációs keretet kell tartalmazzon. A kétfajta keret az első részben különbözik egymástól, de mindkettő tartalmaz hasznos teher részt is. A szinkronizációs keret a műsor leírását tartalmazza. Ez maga a videó kulcskeret vagy közvetlenül előtte kell hogy legyen. A nem-szinkronizációs keret több hasznos terhet szállít, de nem tartalmaz járulékos információkat. Ezeket alacsonyabb sávszélesség esetén alkalmazzák. A hasznos teher minden esetben az aktuális adattípus kódját és magát az adatot tartalmazza. A típuskód függvényében az adat szerkezete beazonosítható, így az adapt struktúra a további csatornák és műsorjellemzők adatait is tartalmazhatja. A hang és a videó adat csomagok egyegy keretben továbbítódnak. Igénytől függően a hang megelőzi vagy követi a videót. Kiegészítő információk (műsor címe, 16:9/4:3 megjelenítési arány, másodlagos hang csatorna, stb.) csatornáinak száma összesen 15 lehet.

8.2. A VoIP hálózat jellemzői

Miután a hang tömörítése és adattá konvertálása megtörtént, az RTP (Real Stream Protocol) segítségével az IP hálózaton megtörténik a jelfolyam továbbítása. VoIP hálózatban úgy a sávszélességet, mint a hálózat késleltetését figyelembe kell venni. A sávszélesség igények kritikusak és nem csak a kiválasztott kodektől függ, hanem az egyes protokollok (IP, UDP, stb.) overhead-jétől is. A késleltetés a jel terjedési sebességétől, a küldő és a fogadó csomópont pufferének kezelési mechanizmusától, valamint a csomagolási késleltetéstől függ.

8.2.1. Sávszélesség követelmények a VoIP hálózatban

A hang párbeszéd IP hálózat feletti működését több tényező befolyásolja. Az alkalmazott kodek sávszélesség igénye a 3…64 kbps tartományban lehet. A hang protokoll adatelem (PDU) leggyakrabban 20 bájtnál rövidebb, míg az L2 (Ethernet) és az L3 (IP) rétegek szignifikáns overhead-et képeznek. Emiatt a valós fizikai sávszélesség igényt nagymértékben az overhead-ek befolyásolják. E probléma egyszerűsítésére különböző megoldásokat vezettek be. Hangaktivitás felismerés (VOD - Voice Activity Detection) segítségével a küldő a csomagolt jelfolyamot megszakítja, ha a lokális analóg forrás jelszint egy megadott küszöbérték alá kerül. Ezáltal a sávszélesség igény közel felére csökken, mivel a humán beszélgetés közben várhatóan a személyek fele ideig a másikat hallgatják. Ez a megoldás viszont körültekintést igényel a ki/bekapcsolási pillanatok meghatározásánál, mivel kieséseket okozhat. Ugyanakkor a beszélgetés közbeni teljes csend is zavaró lehet. Emiatt alkalmazni szokták a komfort zajt, amely a hallgató fél párjánál a hangszóróban lokálisan generált halk fehérzajként jelenik meg. Fejlettebb rendszerek a távoli környező háttérzajt reprodukálják a távoli személy hallgatási időintervallumaiban.

Egy másik megoldás az RTP PDU fejrészének tömörítése. Mivel az RTP PDU fejrészében több információ duplikált vagy redundáns módon jelen van, az útvonal mentén elhelyezkedő routerek a fejrészt tömörítik, így a beszéd számára szükséges sávszélesség lényegesen csökken. A leggyakoribb LAN/MAN technológiai környezetben a szükséges fizikai sávszélesség az 5. táblázat szerint alakul. Az IP/UDP/RTP 40 bájt, az Ethernet pedig 14 bájt overhead-et képez.

Algorithm	*Voice bandw. [kbps]*	*Codec latency [msec]*	*Voice PDU size [B]*	*Voice rate [PDU/sec]*	*L2 PDU size [B]*	*Physical bandw. [kbps]*
G.729	8.0	15.0	20	50	74	29.60
G.711	64.0	1.5	160	50	214	85.60
G.723.1	6.3	37.5	30	26	84	17.47
G.723.2	5.3	37.5	30	22	84	14.78

5. táblázat Algoritmusok és sávszélesség igények

Minden egyes beszédkapcsolat két hívás jelfolyamot, míg a videókapcsolat négy vagy hat egyidejű hívás jelfolyamot jelent.

8.2.2. Késleltetés a VoIP hálózatban

A VoIP rendszerek tervezésénél általánosan elfogadott szabály, hogy a végponttól végpontig terjedő késleltetés 150 ms alatt maradjon. A ma használatos médiák átviteli késleltetése önmagában az emberi fül számára ugyan nem érzékelhető, a kezelési késleltetéssel együttesen azonban már észrevehető torzulást okozhat.

Felhasználói részről a késleltetés tolerancia küszöbe 250 ms. Ennél nagyobb késleltetést elszenvedett hangfolyam interferál a természetes hangfolyammal, így kiolthatják egymást, torzulás érzékelhető. A kezelési késleltetés befolyással van a hagyományos vonalkapcsolt telefonhálózatokra is, de a csomag alapú átvitelnél a pufferelés miatt jelentősége erősen megnő. Ezért a késleltetés tervezésénél ezt 150-200 ms alatt kell tartani.

A G.729 szabvány algoritmus szerinti késleltetése 20 ms körül van, amelynek tervezésénél számításba vették a jövőbeli igényeket is. Egy VoIP termék általánosan 10 ms-onként generál egy keretet, majd párosával helyezi ezeket csomagba, így a késleltetés értéke 20 ms lesz. Csomag alapú hálózat esetén a késleltetés származhat az aktuális csomag kimeneti sorba való helyezéséből, valamint a sor késleltetéséből. Ennek értéke eszközfüggő, optimális esetben nem haladja meg a 30 ms-ot.
A VoIP alkalmazások nemcsak a késleltetésre, hanem annak változására is érzékenyek. Ellentétben a vonalkapcsolt hálózatokkal, a csomagkapcsolt átvitelinél a késleltetés értéke a hálózati forgalomtól függően erősen ingadozhat. A jitter a késleltetésnek rövid időn belüli változása, azaz a csomag várt és valós

érkezési időpontja közötti ingadozás. Az eszközök ezt „playout” pufferekkel kompenzálják, hogy a hang vételében ne
legyenek szakadások. Ez a teljes rendszer késleltetését tovább növeli. A puffer mérete lehet fix nagyságú, illetve bizonyos eszközök esetén adaptív. VoIP esetében a jitter a minőséget legszembetűnőbb módon akadályozó paraméter. Általában a csomagkapcsolt hangátvitelnél a forgalom különböző késleltetésű, és minőségi paramétereket nyújtó rendszereken halad keresztül. Ezek alapvetően gyenge minőséget eredményeznek. Az ilyen alkalmazások általános jellemzője a nagyméretű fogadó oldali puffer, amely általában egy másodperc feletti hanganyag pufferelését teszi lehetővé.

8.2.3. Szolgáltatás minőség a VoIP hálózatban

A csomagkapcsolt hálózatokban a hangminőséget döntően befolyásolja a hálózatra jellemző késleltetés és jitter, így a hálózatok tervezésénél különös figyelmet kell fordítani a QoS paraméterek biztosítására.
További lényeges szempont a hangforgalomnak az adatforgalomtól való védelme, valamint a kritikus adatforgalom védelme a hangforgalom esetleges nagyobb sávszélesség-foglalásával szemben. A hatékony QoS tervezés elemei a megfelelő sávszélesség, a csomagvesztés, a késleltetés, és a jitter. E tényezők megfelelő szintű biztosítása az alábbi leggyakoribb eszközökkel történik.

- *Vezérlési stratégia:* Forgalom limitálás, mely a csomagok eldobását jelenti, amennyiben az adott hálózati eszközök közötti forgalom túllép egy megadott küszöb értéket. Ez megadható az eszközre bemeneti vagy kimeneti oldalon. Tipikus példája a RED (Random Early Detection) és a WRED (Weighted RED). Ezek a technikák beazonosítják azokat a csomagokat, amelyek szükség esetén eldobhatók.

- *Forgalomtervezés:* Egyenletes bemenő és kimenő forgalmú csomagmennyiség alapján biztosítja a pufferelést. A vezérlési stratégiával ellentétben a forgalomtervezés igyekszik elkerülni a csomagok eldobását, ezzel viszont növeli a pufferelésből származó késleltetést és jittert.

- *Híváskezdeményezés kontroll:* Az alkalmazás sávszélesség igényének elutasítását szabályozza. VoIP esetében a hívás számára szükséges sávszélesség lefoglalására használható például az RSVP (Resource Reservation Protocol). Egy H323 gatekeeper korlátozhatja a hívásonként lefoglalható sávszélességet.

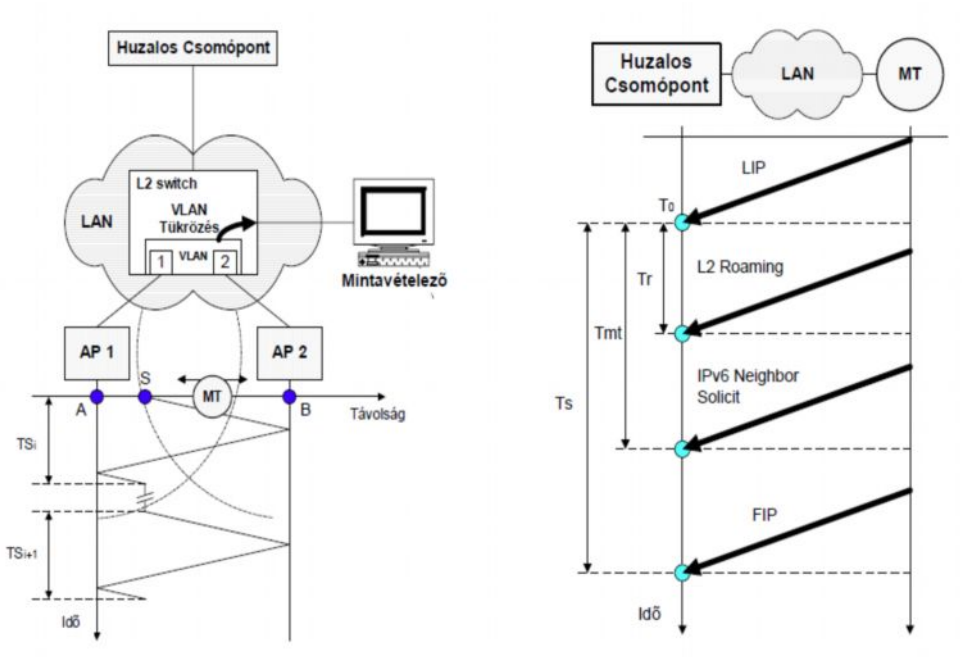
- *Várakozási sorok/ütemezés:* A pufferelés során használható, a csomagok prioritásának felderítésével. Külön sor tartható fenn a késleltetés-érzékeny hangcsomagok, és külön az adatcsomagok számára. VoIP esetében gyakori mechanizmus az IP RTP prioritási sor.

- *Tagging/megjelölés:* Különböző technikák használatosak a speciális kezelést igénylő csomagok megjelölésére. VoIP esetében a csomagok megjelölhetők például az IP precedencia bitekkel (IP fejrész ToS
mezője). A csomagok megjelölés mechanizmusa a hálózat-határokon átnyúló QoS paraméterek megőrzéséhez szükséges.

- *Fragmentálás:* Bizonyos eszközökön engedélyezhető a nagyméretű csomagok további darabolása, mielőtt a kis sávszélességű linken azt továbbítaná. Ez megvédi a hangcsomagokat a nagyméretű adatcsomagok továbbításához szükséges hosszú várakozástól. Így a hangcsomag bekerülhet egy nagyméretű adatcsomag darabjai közé.

8.3. Mérési környezet ismertetése

A mérésekhez olyan eszközparkot használtunk, amelynél úgy a bázisállomások (BS1, BS2), mint a mobil terminál (MT) képes IEEE 802.11a, valamint IEEE 802.11b/g szabványoknak megfelelő mechanizmussal forgalmazni. Ehhez a 35. ábra szerinti beltéri teszt hálózaton gyalogos közlekedés közben a mobil terminal mozgása miatt bekövetkező L2 roaming hatását vizsgáltuk. Az MT 5-6 km/h (1,4-1,7 m/sec) sebességgel haladt a bázisállomásokat összekötő egyenessel párhuzamos irányban oda-vissza. Egy mérési periódus (TSi) alatt az MT a BS1 mikrocellájából L2 romaing hatására átkerült a BS2 mikrocellájába, majd visszafelé haladva újabb L2 romaing hatására visszakerült a BS1 hatáskörébe. Multimédiás szolgáltatásként video streaming és IP telefon alkalmazásokat futtattunk a mobil terminálon. Az MT egy notebook, amelyen Winamp 5.x, illetve SoftPhone szoftverek futottak a streaming (TCP), illetve telefonbeszélgetés (UDP) multimédiás alkalmazásokként.



35. ábra Az MT mozgása a két bázisállomás között

TCP forgalom esetén a huzalos csomópont egy streaming szerver, amiről különböző sávszélességű, NullSoft Video (NSV) szabványú multimédiás műsorokat töltöttünk le (35. ábra). UDP forgalom méréséhez a huzalos és az MT csomóponton SoftPhone telefonszoftvert futtattunk, amelyek között telefonbeszélgetés zajlott. A LAN belsejében elhelyezkedő Phone Center-ben választottuk ki a hangkódolási mechanizmust. A streaming (TCP) műsorok sávszélesség értékei a következők voltak: 80, 150, 300, 500 kbps. A hangkódolási (UDP) mechanizmusok pedig a következők voltak: G.728 (16 kbps), GSM (29 kbps), G.711 (80 kbps), Wideband (272 kbps). Úgy a TCP, mint az UDP forgalmak esetén a bázisállomások Data Retry paraméterét fixen 32-re állítottuk, míg a Beacon periódust a 20ms, 50ms, 100ms értékek között módosítottuk. A WinAmp program fogadó pufferét és a lejátszó puffere is fixen 1000 msec-on állt. Ezek alapján IEEE 802.11 szabványonként TCP-re tizenkét mérést és UDP-re is ugyanennyi mérést végeztünk.

A két bázisállomás (BS1, BS2) egy L2 kapcsoló két portján, ugyanabban az L2 VLAN-ban helyezkedett el. A huzalos hálózaton megjelenő Ethernet kereteket a kapcsoló VLAN-jából egy dedikált fizikai portra történő tükrözéssel juttattuk el a mintavételező géphez, amely TCPDump program segítségével

Libcap formátumú fájlba tárolta azokat. Utólag Ethereal v0.10.14 protokollanalizátor program segítségével elemeztük a tárolt folyamatokat és ezáltal lehetséges volt beazonosítani az alkalmazások minőségét befolyásoló időtartamokat. A bázisállomások által sugárzott rádiós energia IEEE 802.11b/g esetén 100 mW, az IEEE 802.11a esetén, pedig 40 mW volt. A két bázisállomás közötti fizikai távolság 50 méter, az MT rádiós forgalma nyitott autentikáció és titkosítás nélküli volt. Az MT az S pontból indult és a B, majd S, A pontokon újból az S pontba érkezett vissza.

A multimédiás alkalmazások minőségét befolyásoló időtartamok meghatározásához Ethereal/Wireshark analizátorral minden egyes letárolt fájlban beazonosítottuk a 7. ábra szerinti T_0 időpontot. Ez nem más, mint az L2 roaming előtti LIP csomag (Last Important Packet) huzalos csomóponthoz történő beérkezésének időpillanata. Ez tulajdonképpen az MT romaing előtti legutolsó tartalom adata. A T_r időtartam alatt az L2 romaing folyamat játszódik le, aminek részletezése más cikkben található meg[1]. Ennek beazonosítása a romaing keretek új bázisállomáshoz megérkezésével történik. Az MT gépen IPv6 kliens program is fut, amely az IPv4-hez képest rögtön érzékeli a protokoll stack kettes rétegének helyreállását, és azonnal megkezdi a szomszédos csomópontok felfedezését. A Tmt az MT LLC (Logical Link Control) szintű forgalmazás képességének késleltetését jelenti. Az IPv6 kliens e tulajdonságát ahhoz használtuk fel, hogy a határozott romaing kereteket pontosan beazonosíthassuk, mivel az MT S->B->S->A->S beltéri pontok mentén történő haladása közben az épület falain a reflexiók miatt esetenként kettőnél több cellaváltást is tapasztaltunk. A Ts időtartam az MT-n futó multimédia kapcsolat működésének késleltetése. Ez a felhasználó közvetlenül érzékeli és ennek nagy értéke akadozáshoz, illetve a kapcsolat teljes megszakadásához vezethet. Ennek meghatározása a FIP csomag (First Important Packet) beérkezésének beazonosításával történik. TCP, illetve UDP forgalmak esetén a FIP és a FIP csomagokat a 2. Táblázat tartalmazza[J2].

Szállítási réteg	*Fontos csomag*	*Jelentés*
TCP	LIP	L2 roaming előtti utolsó ACK csomag (60 bájt) az MT-től a szerverhez
TCP	FIP	L2 roaming utáni első ACK csomag (60 bájt) az MT-től a szerverhez
UDP	LIP	L2 roaming előtti utolsó UDP csomag az MT-től a huzalos csomóponthoz
UDP	FIP	L2 roaming utáni első UDP csomag az MT-től a huzalos csomóponthoz

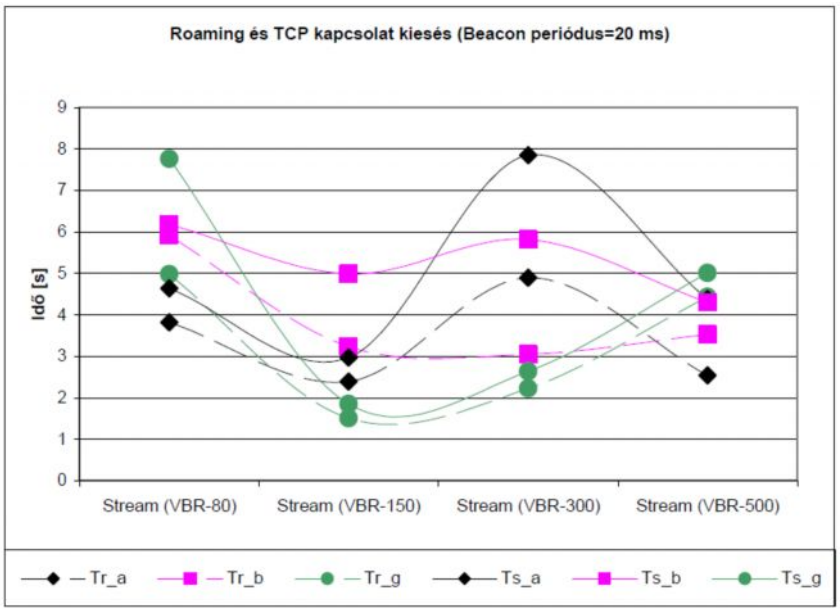
12. táblázat Megkülönböztetett csomagok

Tr időtartam alatt az L2 roaming folyamat játszódik le, aminek részletezését más cikkben mutattuk be[1]. A Tr időtartam beazonosítása a roaming keretek új bázisállomáshoz megérkezésének érzékelésével történt. Az MT gépen IPv6 kliens program is fut, amely az IPv4-hez képest rögtön érzékeli a protokoll stack kettes rétegének helyreállását, és azonnal megkezdi a szomszédos csomópontok felfedezését. A Tmt időtartam az MT LLC (Logical Link Control) szintű forgalmazás képességének késleltetését jelenti. Az IPv6 kliens e tulajdonságát ahhoz használtuk fel, hogy a határozott roaming során küldött kereteket pontosan beazonosíthassuk, mivel az MT S->B->S->A->S beltéri pontok mentén történő haladása közben az épület falain bekövetkező reflexiók miatt esetenként kettőnél több cellaváltást is tapasztaltunk. A Ts időtartam az MT-n futó multimédia kapcsolat működésének késleltetése. Ezt a felhasználó közvetlenül érzékeli és ennek nagy értéke a szolgáltatás akadozásához, illetve a kapcsolat teljes megszakadásához vezethet. Ennek meghatározása a FIP csomag (First Important Packet) beérkezésének beazonosításával történt.

8.4. Új eredmények és értelmezésük

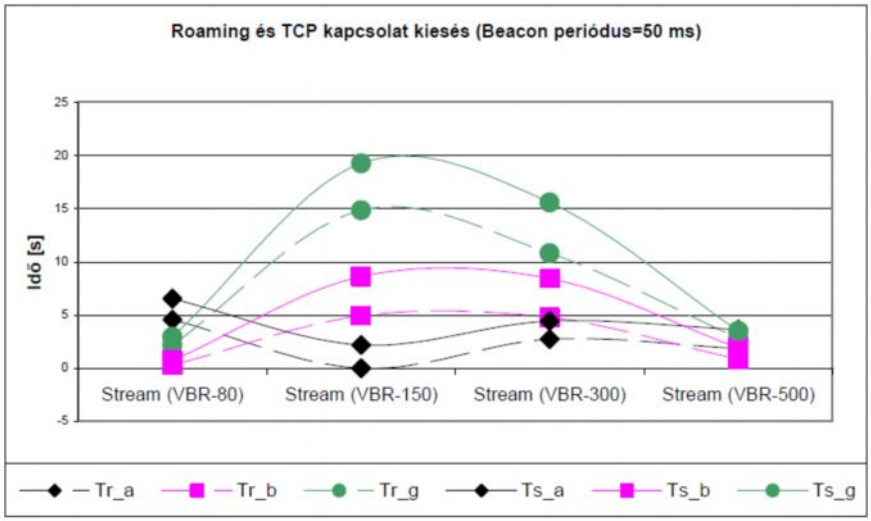
A mérési eredmények összehasonlítása és elemzése fontos következtetések levonására ad lehetőséget. A különböző IEEE 802.11 szabványok eltérő módon viselkednek beltéri környezetben végrehajtott cellaváltások esetén[7].

3. Tézis: A roaming folyamat lejátszódása nagymértékben függ a bázisállomáson beállított beacon periódus (Tb) időtől. Ez a periódus egyben konfigurációs paraméter is[8]. A terminál a beacon-ben továbbított jelzés alapján megtanulja a bázisállomás periódusát[9]. Amennyiben az MT nyolc periódus ideig nem kap beacon-t, kezdetét veszi a roaming folyamat[1]. A beérkező beacon keretek folyamatos figyelésével az MT érzékeli a vezeték nélküli kapcsolat minőségének romlását és cellaváltást kezdeményez. A TCP forgalom mért értékeinek grafikonjait a 36., 37., 38. ábra, az UDP forgalomra vonatkozó grafikonokat, pedig a 39., 40., 41. ábra mutatja[J2].



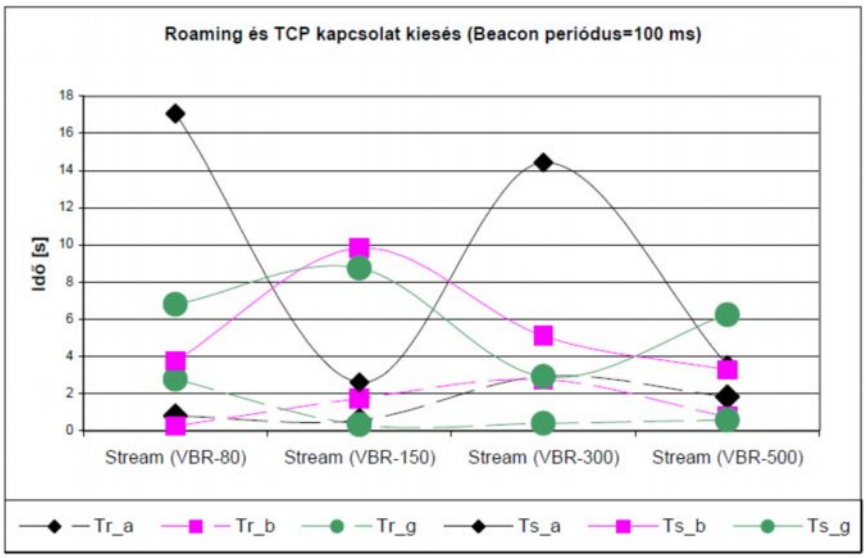
36. ábra Romaing és TCP kapcsolat kiesés (beacon=20ms)

Ha beacon periódust az alapértelmezett 100 ms értékről 50 ms, majd 20 ms-ra csökkentjük, akkor az MT gyakrabban érzékeli a jel/zaj viszony változását, így mozgás közben érzékenyebb lesz a környezeti viszonyok változására. Ilyenkor minden egyes streaming technológia esetén a cellaváltási idő 0,5-2,8 sec értékről előbb 0,1-14,9 sec értékre nő, majd 1,5-5,9 sec értékre csökken, a TCP kapcsolat kiesése pedig 2,5-17 sec értékről előbb 2,4-19,8 sec értékre nő, majd 1,8-7,9 sec értékre csökken. Tehát a Tb=20 msec periódusidő jobb, mint a 100 msec érték. Ez hasznos konfigurálási jelenségnek számít. A Tb nagyon kis értékre vétele sem lehet jó a gyakorlatban, mivel a beltéri környezetben a többútas terjedés miatt a túlzott érzékenység gyakori cellaváltást okoz, ami a TCP kapcsolat gyakori kiesését jelenti.



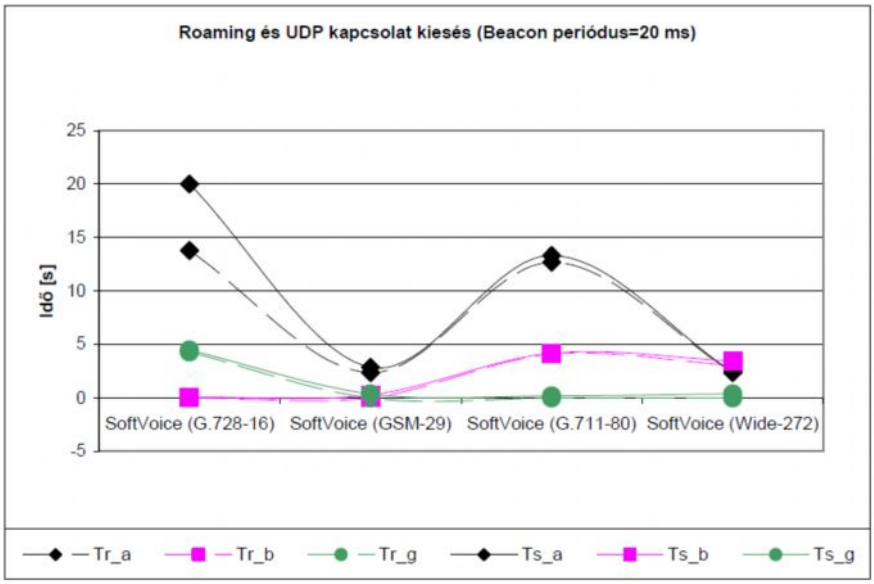
37. ábra Romaing és TCP kapcsolat kiesés (beacon=50ms)

IP telefon kapcsolat esetén adott hangkódolási technikánál a cellaváltási időre a beacon periódus csökkentése 100 msec értékről 50 msec, majd 20 msec-re a cellaváltási időt a 0,1-44,5 sec értékről 0,1-12,5 sec értékekre csökkenti. Az UDP kapcsolat kiesés a 0,2-49,8 sec értékről a 0,2-19,9 sec értékekre csökken. Ez is a Tb=20 msec érték előnyét jelzi.



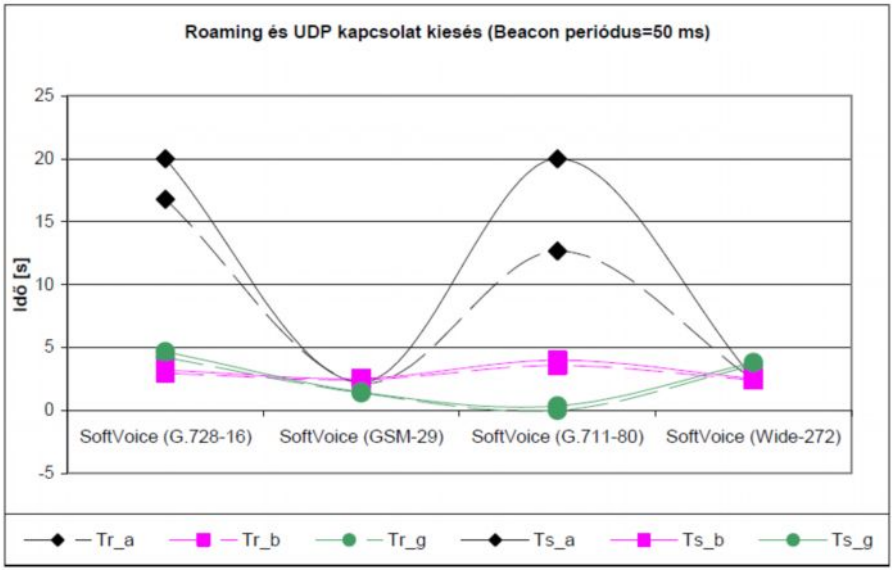
38. ábra Romaing és TCP kapcsolat kiesés (beacon=100ms)

Épületen belül az IEEE 802.11 technológiák különbözőképpen reagálnak a beacon periódusra. Streaming forgalom esetén az IEEE 802.11a hosszabb ideig állítja vissza a kapcsolatot. Utána az IEEE 802.11b következik, és legelőnyösebb tulajdonságokkal az IEEE 802.11g rendelkezik beltéri cellaváltás esetén.



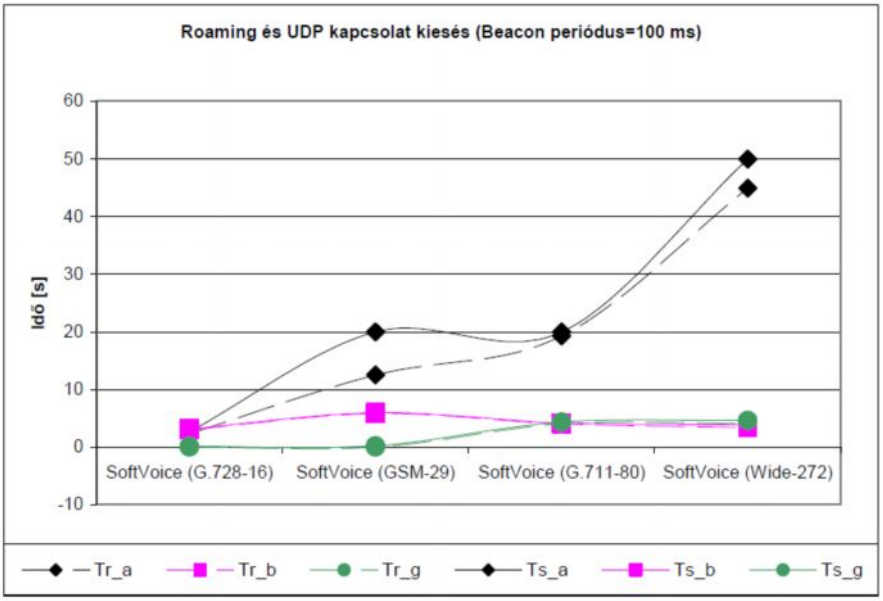
39. ábraRoaming és UDP kapcsolat kiesés (beacon=20ms)

IP telefon szolgáltatásnál az IEEE 802.11a nagyon nagy késleltetéseket produkál, nagy sávszélességű hangkapcsolat esetén le is szakad a szolgáltatás. Az IEEE 802.11g a legjobb reakcióidőt biztosítja, így a szolgáltatás kiesése 4 sec alatti. Ez még elviselhető ritka esemény lehet mozgó IP telefonos környezetben, ha a felhasználók erről előzetes értesüléssel rendelkeznek.



40. ábra Roaming és UDP kapcsolat kiesés (beacon=50ms)

A streaming sávszélesség igényétől függően a műsor kiesési idő is eltérő viselkedést mutat. A 150 kbps-os NSV műsor Tb=20 msec esetén a legkevésbé függ a rádiós technológiától, viszont Tb=50 msec esetén éppen a 150 kbps-os NSV műsor függ leginkább a rádiós technológiától. Tb≤50 msec esetén a 80 kbps-os és az 500 kbps-os NSV műsorok kevésbé függnek a rádiós technológiától.



41. ábra Roaming és UDP kapcsolat kiesés (beacon=100ms)

Az IP telefon kapcsolat Tb≤50 msec esetén a GSM hangkódolásnál mutatja a legkisebb kiesést. Ez a GSM technológia mobil viszonyokra optimalizált tulajdonságából adódik. Annak ellenére, hogy a GSM minőségben gyengébb, mint a G.728, mégis jobban illeszkedik a cellaváltás okozta környezetváltozáshoz. Tb≤50 msec esetén a G.711 nagyon függ a rádiós technológia cellaváltási mechanizmusától. Ez a PCM hagyományos huzalos környezetre kialakított tulajdonságából adódik[J2].

- Cellaváltás befejezése után a streaming kapcsolat folytatása nagy beacon periódusidő esetén későn, 4-11 sec késleltetéssel történik. Ezzel ellentétben Tb≤50 msec esetén az új cellába váltás után a TCP kapcsolat 3 sec késleltetés után folytatódik. Ez az adott technológia Ts-Tr időkülönbség értékekből figyelhető meg.
- IP telefon esetén a cellaváltás utáni UDP forgalom folytatásának késleltetése IEEE 802.11b/g radios technológiák esetén minden hangkódolási technika esetén 0,5 sec alatt van. Az IEEE 802.11a viszont az MT új cellába érkezése után még az UDP forgalom folytatását is jelentősen (3-7 sec) késlelteti.
- Az IEEE 802.11a rádiós technológia a streaming számára Tb=50 msec értékre mutatja az összességében legelőnyösebb tulajdonságait. Az IEEE

802.11g pedig a Tb=20 msec értékre képes legelőnyösebb adatkapcsolati szolgáltatást nyújtani a streaming részére.

- Az IP telefon számára hangkódolási technikától függetlenül bármilyen beacon periódusra az IEEE 802.11g rádiós technológia a legjobb, ezt követi az IEEE 802.11b, majd a legkedvezőtlenebb viselkedést az IEEE 802.11a mutatja.
- Az IP telefon esetén a cellaváltásból származó adatkapcsolati szintű kimaradás miatt a hangkódolási technikák rugalmassági sorrendje csökkenő sorrendben a következő: GSM, Wideband, G.711, G,728.

4. Tézis: Az IEEE 802.11a szabvány által mutatott kedvezőtlenebb mérési eredmények magyarázata, hogy az 5Ghz-es tartományban több csatorna áll rendelkezésre (Európában 19 nem átfedő csatorna), mint a 2,4Ghz-es sávban (3 nem átfedő csatorna), így cellaváltáskor a szkennelési idő (channel probe) is természetszerűleg megnövekszik. Minthogy a kliens a szkennelést csatornaszám és frekvencia tekintetében növekvő sorrendben hajtja végre az 1. csatornától egészen a keresett csatornáig, így két szomszédos rádiós cella esetén a cellaváltási sebességet optimalizálni lehet azáltal, hogy két szomszédos nem átfedő csatornát alkalmazunk. A szabvány meghatározza a csatorna szkennelési időt:

$$N_{ch} * T_{min} <= t <= N_{ch} * T_{max},$$

ahol N_{ch} a csatornák száma, T_{min} az a legkisebb idő, amennyit a kliensnek várakoznia kell a probe response válaszcsomagra, T_{max} az a maximális idő amennyit várakoznia kell a kap probe response-ra, t a teljes effektív szkennelési idő.

A gyakorlatban a legtöbb gyártó 30-40ms tartományban állítja be a T_{max}-ot. A legkedvezőtlenebb esetben ez akár 570-760ms teljes szkennelési időt eredményezhet.

A 802.11a szabvány egyik hátránya, hogy a bázisállomások automatikusan választanak csatornát, így nincs mód a roaming szempontból optimális csatornák explicit kiválasztására.

9. Összefoglalás

Ebben az értekezésben a végponttól végpontig terjedő IP alapú adatátvitel minőségében és teljesítményében alapvető szerepet játszó protokollokat vizsgáltuk különböző L1/2-es hálózati technológiai környezetben. Célunk az volt, hogy rávilágítsunk azokra a lehetőségekre, melyek megoldást jelenthetnek a TCP torlódásvezerlés okozta alacsony átviteli teljesítményre.

Aszimmetrikus kapcsolatok

A TCP teljesítményét egyértelműen meghatározza a nyugták beérkezési ideje. Az erősen torlódott felfeléirányú kapcsolaton a nyugták viszont csak késleltetéssel továbbíthatóak. Ebben a fázisban a lefelé irányú TCP folyam átviteli sebessége a rendelkezésre álló fizikai sávszélesség 70%-a körüli értékre csökken, az alkalmazott torlódásvezérlés függvényében. A kliens irányából újabb TCP folyamot indítottunk, még erőteljesebb torlódási jelenséget generálva a felfeléirányú kapcsolaton, ami további átviteli-teljesítmény csökkenést eredményezett a lefeléirányú TCP folyamon (3. fázis). Az effektív átviteli ráta ekkor 40% körüli értéket vett fel. Ebben a helyzetben az erősen torlódott, eredendően is alacsony sávszélességű felfelé irány következtében a rendelkezésre álló lefeléirányú sávszélességet a TCP nem képes hatékonyan kihasználni. Így kimutattuk, hogy a felfelé irány alacsony sávszélessége közvetlen hatással van a lefeléirányú TCP folyam átviteli teljesítményre.

1. Tézis: Rámutattam, hogy a felfelé irányú kapcsolati terheltség az alkalmazott torlódásvezérlő algoritmustól függetlenül közvetlen hatással van a lefelé irányú TCP folyam teljesítményére, minthogy uplink torlódás hatására a TCP önszabályzó mechanizmusa lecsökkenti az effektív ablakméretet. További problémát okoz a nyugták érkezési időközeinek burst-ös változása *(ACK compression)* az uplink-en, ami torlódást okoz a lefelé irányon. Ebben a helyzetben a TCP nem képes hatékonyan kihasználni a fizikai downlink sávszélességet[J4].

Nagysebességű, magas késleltetésű kapcsolatok

Ebben a tesztsorozatban célom az volt, hogy a nagysebességű kapcsolat B x D szorzata alapján az alapértelmezett TCP kernel-változók hangolásával javítsam a TCP kapcsolat átviteli karakterisztikáját.

2. Tézis: Nagysebességű, magas késleltetésű hálózatokon a TCP-t hangolni kell az optimális átviteli teljesítmény eléréséhez. A B x D szorzat, az egyidejű TCP kapcsolatok száma, valamint a csomagvesztési arány ismeretét felhasználva a TCP kernel-változók megfelelő hangolásával jelentős, egyes esetekben közel 200%-os átviteli teljesítménynövekedés érhető el. Ugyanakkor bármely TCP variánst is néztem, az általam végzett mérések során alkalmazott relatíve alacsony (~4,4ms) késleltetésű WAN környezetben a torlódásvezérlő algoritmus optimálisra hangolt kernel válozókkal sem volt képes az 1 Gbit-es kapcsolat fizikai sávszélességéhez közeli átviteli teljesítményt nyújtani [C7].

Vezeték nélküli védett hálózatok

Hangsúlyozottan az adatkapcsolati szintű roaming eseményre fokuszáltunk. Azt láttuk, hogy a 802.11a szabvány mobilitás tekintetében nem igazolta hatékonyságát. Nyilvánvalóan a hosszabb mérési intervallumokon túl a roaming esemény a TCP kapcsolatok lebontását is eredményezte számos esetben, annak ellenére, hogy a 802.11a a kevésbe terhelt, tehát kevésbe zajos 5.4GHz-es tartományban üzemel. Mindegyik biztonsági protokoll kombináció többletterhelést jelent a roaming időre. Következésképpen az alkalmazási rétegben érzékelhető QoS paraméterek minőségét is csökkentik. Bármely roaming szakaszt is nézzük, jelentős különbségek adódnak a hitelesítési mechanizmusok (MSCHAP, GTC) és a vezetéknélküli technológiák között is (IEEE 802.11a/b/g). Az LLC (Logical Link Control) aktivitás ideje valójában az alkalmazott PEAP verziótól (vagyis a tunel-en belül használt hitelesítéstől) függ. Ebben a tekintetben az L2-es roaming minden esetben gyorsabban ment végbe MSCHAP belső hitelesítéssel. Ennek ellenére a hitelesítő és titkosító mechanizmusok közötti eltérések jóval kisebb nagyságrendűek, mint a szállítási rétegben tapasztalható késleltetés, mely nagyságrendileg a másodperces tartományban mozog.

Mobil WiFi hálózatok

A mobil WiFi környezetben elérhető alkalmazási rétegbeli szolgáltatás minőség a komplex roaming folyamattól függ, ahol az összteljesítmény a protokoll stack számos komponensének függvénye, hangsúlyozva az alkalmazott TCP torlódásvezérlő mechanizmus jelentőségét.

Minél hosszabb a roaming periódus, annál nagyobb számú TCP csomagot kell újraküldeni, ami ennek megfelelően rontja a TCP átviteli teljesítményét és a szolgáltatásminőséget[J1].

A mobil kliens bázisálomásokhoz viszonyított relatív sebessége és a roaming végrehajtásának kölcsönhatása jelentősen befolyásolja a TCP kapcsolatokat, miközben kevésbé hat az UDP átvitelre[5,6]. Az összehasonlító mérésekből statisztikai módszerekkel nyert eredmények lehetővé teszik, hogy valós képet kapjunk az IPv4 és az IPv6 mobil átvitel esetén tanusított viselkedésére vonatkozóan, valamint választ kaphatunk arra a kérdésre, hogy valóban magasabb minőségű mobil adatátvitelt eredményez-e az IPv6 protokoll vezeték nélküli adatkapcsolati réteg fölött elődjéhez, az IPv4-hez képest. A hagyományos elektronikus alkalmazások az IPv4 protokoll „best effort" jellege miatt lasúbb átvitelt biztosítanak mobil WiFi környezetben, míg az IPv6 protokoll az alsóbb rétegekhez történő gyors adaptáció miatt hatékony átvitelt képes biztosítani.

Az időérzékeny alkalmazások (IP telefon, videokonferencia, stb.) az IPv4 protokoll QoS korlátai miatt mobil WiFi környezetben nagy kieséseket szenvednek, így az eredményül kapott szolgáltatás-minőség elfogadhatatlan. Az IPv6 gyors adaptációja miatt a kiesések kisebbek, ezért a jelenlegi mobil WiFi környezetben fast roaming esetén közel elfogadható szolgáltatási minőséget nyújt az infokommunikációs alkalmazások számára[4].

Multimédia alkalmazások mobil 802.11a/b/g hálózatokon

A különböző IEEE 802.11 szabványok eltérő módon viselkednek beltéri környezetben végrehajtott cellaváltások esetén[7].

3. Tézis: Kimutattuk, hogy a roaming folyamat lejátszódása nagymértékben függ a bázisállomáson beállított beacon periódus (Tb) időtől. Ha beacon periódust az alapértelmezett 100 ms értékről 50 ms, majd 20 ms-ra csökkentjük, akkor az MT gyakrabban érzékeli a jel/zaj viszony változását, így mozgás közben érzékenyebb lesz a környezeti viszonyok változására[J2].

Vizsgálataink összegzéseként megállapítjuk, hogy a legoptimálisabb TCP átviteli teljesítményt akkor kapjuk, ha az adott hálózati paraméterek, valamint forgalom-karakterisztika ismeretében választjuk ki a megfelelő torlódásvezérlő mechanizmust, valamint a kernel-változók értékeit megfelelő eljárással összehangoljuk a forgalom karakterisztikája alapján: sávszélesség, késleltetés, egyidejű TCP kapcsolatok száma, csomagvesztés, forgalom jellege, teljesítmény és méltányosság, valamint beacon periódus.

A közelmúltban már megjelentek bizonyos operációs rendszereken olyan TCP változatok, mely képesek a TCP kernel-változók dinamikus szabályozására a kapcsolati paraméterek függvényében. A korszerű mechanizmusok minden kiépülő TCP kapcsolathoz karakterisztikájának megfelelő (útvonal sávszélessége, késleltetés, konkurens kapcsolatok, forgalom jellege) egyéni puffer méreteket, és válaszfüggvény paramétereket állítanak be, bizonyos esetekben kettős torlódásvezérlést alkalmaznak.

4. Tézis: Kimutattuk, hogy az L2-es roaming esemény IEEE 802.11a szabványú hálózaton jelentősen hosszabb időt vesz igénybe, mint a 802.11b/g esetén. A valósidejű multimédia alkalmazások számára a szabvány 802.11a technológia roaming teljesítménye elfogadhatatlan szolgáltatásminőséget nyújt cellaváltáskor.

A kedvezőtlenebb mérési eredmények magyarázata, hogy az 5Ghz-es tartományban több csatorna áll rendelkezésre (Európában 19 nem átfedő csatorna), mint a 2,4Ghz-es sávban (3 nem átfedő csatorna), így cellaváltáskor a szkennelési idő (channel probe) is természetszerűleg megnövekszik. Minthogy a kliens a szkennelést csatornaszám és frekvencia tekintetében növekvő sorrendben hajtja végre az 1. csatornától egészen a keresett csatornáig, így két szomszédos rádiós cella esetén a cellaváltási sebességet optimalizálni lehet azáltal, hogy két szomszédos nem átfedő csatornát alkalmazunk. A szabvány meghatározza a csatorna szkennelési időt:

$$N_{ch} * T_{min} <= t <= N_{ch} * T_{max},$$

ahol N_{ch} a csatornák száma, T_{min} az a legkisebb idő, amennyit a kliensnek várakoznia kell a probe response válaszcsomagra, T_{max} az a maximális idő amennyit várakoznia kell a kap probe response-ra, t a teljes effektív szkennelési idő.

A gyakorlatban a legtöbb gyártó 30-40ms tartományban állítja be a T_{max}-ot, ez a timeout tartomány nem alkalmas valósidejű multimédia alkalmazások számára, mivel a legkedvezőtlenebb esetben T_{max} akár 570-760ms teljes szkennelési időt eredményezhet.

Summary

In this dissertation we have analysed networking protocols that play important roles in the quality and performance issues of end-to-end IP-based data transmission over some novel layer 1 and 2 networking technologies. Our aim was to present possible solutions that could eliminate the poor transmission performance of TCP congestion control.

Asymmetric connections

The performance of TCP is determined by the arrival time of acknowledgements. Acknowledgements could be transmitted with high latency on a heavily congested narrow band uplink. In our first measurement phase the transfer rate of the TCP stream decreased and then fluctuated around 70% of the available net bandwidth depending on the applied congestion control mechanism. After the first stream reached its steady state another TCP stream has been initiated from the client generating a more serious congestion event on the uplink that drove to a further transfer rate degradation (see third phase). Effective rate has fallen down to approx. 40% and around due to the delayed acknowledgements on the fully congested low bandwidth uplink. In this case TCP was not able to utilize the available downlink bandwidth effectively.

Thesis #1: I pointed out how uplink load impacts the performance of downlink TCP stream irrespectively of the applied congestion control algorithm, since TCP's self-clocking mechanism decreases the effective window upon uplink congestion. Further issue rises due to the bursty arrival time of the ACKs (compression) on the uplink that drives to a congestion event on the downlink. In this situation TCP is not able to effectively use the available bandwidth on the downlink[J4].

Narrow band uplink shall continue to be a serious bottleneck for a well-defined type of network traffic on asymmetric connections.

High bandwidth-delay product connections

With this series of tests we intended to enhance the transmission characteristics of TCP by tuning the default TCP kernel variables based on the B x D product of the connection.

Thesis #2: In order to get the optimal end-to-end transmission performance we need to adjust TCP parameters on high BDP networks. Using the B x D product, number of concurrent TCP sessions and packet loss ratio we can adjust kernel level TCP variables achieving a significant increase (200 percent in some cases) in the transmission performance. However, none of TCP variants I observed was able to get close to the available physical bandwidth of the 1Gbps connection even with optimized kernel variables on the relatively low-latency (~4,4msec) long distance network used in my measurements[C7].

Secured WiFi networks

We have investigated the performance of IEEE802.11i-based security suites on IEEE802.11a/b/g mobile WiFi systems. We have focused on L2 roaming events. We found that the 802.11a standard cannot prove its efficiency in terms of mobility. Obviously, apart from presenting longer intervals for all measured phases, roaming events produced disconnection of the TCP session in several cases. However, 802.11a operates at 5.4GHz frequency range that is less congested. All of the measured security protocol combinations added their overhead to roaming time. Therefore, the QoS parameters experienced at the application layer fell down. If we study any roaming phase, significant differences arise between authentication mechanisms (MSCHAP, GTC) and also between WiFi technologies (IEEE 802.11a/b/g). Time of LLC activity actually depends on the applied PEAP version. At this point L2 roaming was faster with MSCHAP in all cases. However, differencies between the performance of authentication and encryption mechanisms are smaller of order compared to the transport layer latency that is in the second time domain.

Mobil WiFi networks

Applicatons layer QoS in mobile WiFi environment depends on a complex roaming process where overall performance is function of several component of protocol stack especially of the congestion control mechanism applied.

The longer the roaming phase the larger number of TCP segments have to be resend that negatively affects the TCP transmission and the quality of service[J1].

TCP connections are significantly, while UDPs are less affected by the interaction between the mobile station's relative speed and the roaming execution[5,6]. The results gained statistically from the comparative measurements provide us a practical review about the behavior of IPv4 and IPv6 protocols in mobile environment. Furthermore we concluded that the performance of IPv6 over wireless data link layer is really higher compared its predecessor IPv4. The conventional applications provide slower transmission over mobile links due to the best effort nature of IP, while IPv6 assures effective data transfer because of its quick adaptation to lower layers.
Time sensitive applications (IP phone, video conference) suffer significant dropouts due to the limitations of IPv4's QoS in mobile WiFi environment therefore the provided quality is unacceptable. The quick adaptation of IPv6 decreases the interval of dropouts [4].

Multimedia applications on mobile 802.11a/b/g networks

Different IEEE 802.11 technologies have dissimilar behaviour during L2 roaming event in indoor environment[7].

Thesis #3: We pointed out that roaming process heavily depends on the beacon period (Tb) set on the access points. When the beacon period decreases from the default 100ms to 50ms and 20ms then mobile client is able to sense the signal-to-noise ratio more ofter therefore it will be more sensitive to alteration in the RF environment[J2].

Summarizing our investigations we can establish that optimal performance of TCP transmission could be reached by applying congestion control mechanism based on our knowledge about the given network' parameters and traffic

characteristics. Furthermore we should adjust kernel variables according to the bandwidth, delay, concurrent TCP streams, packed loss, traffic shape, performance, fairness and beacon period .

Recently such TCP variants had been released with some Operating Systems that are capable of adjusting TCP kernel variables according to the network parameters. Novel mechanisms set up buffer sizes and response functions that adapt to the characteristics of the TCP connection.

Thesis #4: We presented that L2 roaming process takes significantly more time on IEEE 802.11a network than on 802.11b/g. Roaming performance of 802.11a technology provided an unacceptable quality of service for real time multimedia applications.

Explanation of the worse measurement values is the following: more non-overlapping channels (19 in EU) are available on the 5Ghz frequency range than on the 2.4Ghz. Channel probing time subsequently increases at radio cell changes. Since a wireless client scans radio channels starting from channel 1 to the channel of the AP, roaming time could be optimized by selecting the lowest non-overlapping consecutive channels for neighboring APs. The standard defines the channel probing time values as following:

$$N_{ch} * T_{min} <= t <= N_{ch} * T_{max},$$

where N_{ch} is the number of channels, T_{min} is the smalles time period that the client has to wait for probe response, T_{max} is the maximum time value waiting for, t is the total channel probing interval.

In practice most of the hardware vendors set T_{max} up on their devices in the 30-40ms range, however this timeout interval is not adequate for real time multimedia applications, since T_{max} in the worst case could result in an overall scanning time of 570-760ms.

Irodalomjegyzék

[1] Allman, M. – Paxson, V. – Stevens, W.: *"TCP congestion control"*, RFC 2581, http://www.faqs.org/rfcs/rfc2581.html

[2] Postel, J., *"Transmission Control Protocol"*, RFC 793, September 1981.

[3] Jacobson, V., Braden, R., and Borman, D., *"TCP Extensions for High Performance"*, RFC 1323, May 1992.

[4] Jacobson V., *"Congestion Avoidance and Control"*, ACM Computer Communication Review, Vol. 18, No. 4, August 1988.

[5] Mathis, M., Madavi, J., Floyd, S., and Romanow, A., *"TCP Selective Acknowledgement Options, "* RFC 2018, October 1996.

[6] Stevens, W. R., *"TCP/IP Illustrated"*, Volume 1, Addison-Wesley, 1994.

[7] Geoff Huston, *"TCP Performance"*, The Internet Protocol Journal - Volume 3, No. 2

[8] Jeonghoon Mo, Richard J. La, Venkat Anantharam, and Jean Walrand: *"Analysis and Comparison of TCP Reno and Vegas"*, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley

[9] TCP Westwood reference pages: http://www.cs.ucla.edu/NRL/hpi/tcpw/

[10] TCP BIC/Cubic reference pages: http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/index_files/Page703.htm

[11] S. Floyd, *"High Speed TCP"*, IETF RFC 3649, http://www.faqs.org/rfcs/rfc3649.html

[12] Scalable TCP: *"Improving Performance in Highspeed Wide Area Networks"*, Tom Kelly, CERN/University of Cambridge

[13] IEEE, Part 11: *"Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications"*. IEEE Std 802.11-1999, 1999.

[14] M. Heusse, F. Rousseu, G. Berger-Sabbatel, and A. Duda: *"Performance Anomaly of 802.11b"*, in Proc. IEEE INFOCOM'03, pp. 836-843, 2003.

[15] Microsoft TechNet, , *"Introduction to Mobile IPv6"*, The Cable Guy – Sept-2004

http://www.microsoft.com/technet/community/columns/cableguy/cg0904.mspx

[16] Charles E. Perkins Sun Microsystems *"Nomadicity: How Mobility Will Affect the Protocol Stack"*,

[17] Microsoft Corporation: *"Understanding Mobile IPv6"*, http://www.microsoft.com/downloads/details.aspx?FamilyID=f85dd3f2-802b-4ea3-8148-6cde835c8921&displaylang=en

[18] G. Xylomenos, G. Polyzos, P. Mahonen, and M. Saaranen. *"TCP performance issues over wireless links"*. IEEE Communications Magazine, 39(4):52–58, 2001

[19] Jonathan Leary, Pejman Roshan. "*Wireless LAN Fundamentals: Mobility",* CiscoPress, Jan. 2004.

[20] W. Haitao, P. Yong, L. Keping, C. Shiduan, and M. Jian. *"Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement"*. In Proc. IEEE INFOCOM '02, pages 599–607, 2002.

[21] S. Choi, K. Park, and C. Kim, *"On the Performance Characteristics of WLANs: Revisited,"* in Proc. ACM SIGMETRICS'05, pp. 97-108, 2005.

[22] Ye Tian, Kai Xu, Nirwan Ansari: *"TCP in Wireless Environments: Problems and Solutions"*, IEEE Radio Communications, March 2005.

[23] Zoltán Gál, György Terdik: *"Multifractal Study of Wireless and Wireline Datanetworks"*, 8th International Conference on Advances in Communications and Control, Telecommunications/Signal Processing - Proceedings, Crete, Greece, 25-29 June 2001.

[24] Cisco Systems, Inc.: *"Cisco Fast Secure Roaming"*

[25] V. Joel, *"Exploding the myth of WLAN performance"*, Telephony Online, October 2004.

[26] Ramya Raghavendra, Elizabeth M. Belding, Konstantina Papagiannaki, Kevin C. Almeroth: *"Understanding Handoffs in Large IEEE 802.11 WirelessNetworks"*, Department of Computer Science, University of California, Santa Barbara, Intel Research, Pittsburgh

Orosz Péter publikációi

Nemzetközi folyóirat cikkek

[J1] *Evaluation of IPv6 Services in Mobile WiFi Environment* – Zoltán Gál, Péter Orosz, Andrea Karsai. Selected Papers of Info-Communications Technology, Volume LX., (2005) pp 47-54.

[J2] *Effect of WiFi Systems on Multimedia Applications* – Zoltán Gál, Andrea Karsai, Péter Orosz. Info-communications-technology Volume LXII. 2007 (2007/1) Selected papers pp 8-14.

[J3] *Performance Evaluation of Centralized IEEE802.11i-based Security Suites on Mobile WiFi Networks* – Péter Orosz, János Sztrik, Seokjun Lee, Youngjin Oh, Chesoong Kim, Telecommunications Review Vol. 17 No. 6, (2007/12) pp 1133-1143.

[J4] *Dynamics and Congestion Control of Alternative TCP Variants on Asymmetric Lines* – Péter Orosz, János Sztrik, Chesoong Kim. ISAST Transactions on Communications and Networking, No. 1 Vol. 2, (2008) pp 71-74.

Előadások, konferencia cikkek

[C1] *Vastag kliensek menedzsmentje Tivoli környezetben* / *Management of Thick Clients in IBM Tivoli Environment* – Orosz Péter, Gál Zoltán. Networkshop 2004 Conference, Győr, Hungary.

[C2] *IP kapcsolatok elemzése mobil WiFi környezetben* – Gál Zoltán, Orosz Péter, Karsai Andrea. Networkshop 2005 Conference, Szeged, Hungary.

[C3] *Központosított EAP alapú hitelesítés vezték nélküli hálózatokban* / *Centralized EAP Based Authentication for Wireless Networks* – Orosz Péter, Sztrik János, Chesoong Kim. Informatics in Higher Education 2005 Conference, Debrecen, Hungary.

[C4] *WiFi rendszeren működő multimédiás alkalmazások elemzése* – Gál Zoltán, Karsai Andrea, Orosz Péter. Networkshop 2006 Conference, Miskolc, Hungary.

[C5] *Adatbiztonság elemzése mobil WiFi környezetben* / *Security Analysis in Mobile WiFi Environment* – Orosz Péter, Gál Zoltán, Karsai Andrea. Networkshop 2006 Conference, Miskolc, Hungary.

[C6] *TCP dynamics and congestion control on asymmetric lines* – Orosz Péter, Sztrik János, Che Soong Kim. ICAI ’07 Nemzetközi konferencia és Proceedings, Eger, Hungary.

[C7] *Alternatív TCP variánsok és torlódásvezérlő mechanizmusok vizsgálata magas késleltetésű, nagy sávszélességű hálózatokon / Observation of alternative TCP variants and congestion control mechanisms on high bandwidth delay networks* – Orosz Péter. Networkshop 2007, Eger, Hungary.

[C8] *Alternatív TCP torlódásvezérlő mechanizmusok vizsgálata magas késleltetésű, nagy sávszélességű hálózatokon / Observation of alternative TCP congestion control mechanisms on high bandwidth delay networks* – Orosz Péter. Informatika a felsőoktatásban 2008, Debrecen, Hungary.