

# Választott CCN megoldások implementálása

Németh Felicián

## Kivonat

A tartalomközpontú hálózatok (CCN) az Internet Protokoll folyamatos toldozgatásával szemben egy, az alapoktól újragondolt, egységes szervező elem mentén létrejött újszerű hálózati megközelítést képvisel. A CCN általános felépítésének bemutatása után a tanulmány a többesadás megvalósíthatóságát ismerteti, majd a kiválasztott CCN komponenseket veszi sorra. A komponensekből épített, emulált teszthálózat segítségével végzett teljesítménymérések eredményeiből leszűrhető, hogy a CCN hálózatok jó hatásfokkal alkalmazhatók médiatartalmak átvitelére mobil környezetben és többesadós elrendezés esetén is.

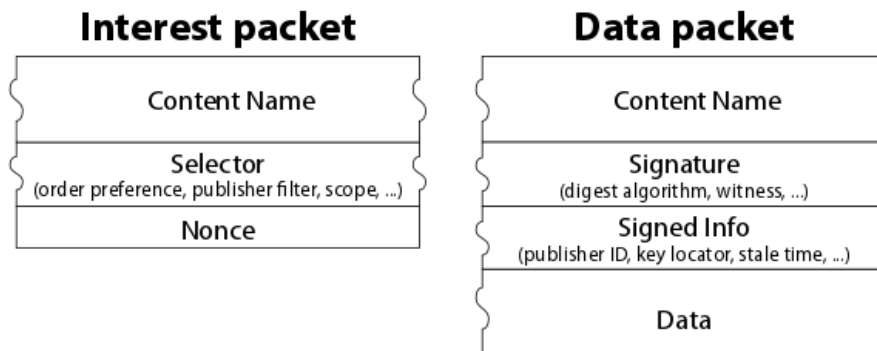
## 1. Bevezetés

A mostani Internet technológiai alapját adó IP protokollsalád folyamatosan fejlődött és fejlődik. A TCP protokoll első változataikor például még nem volt szempont, hogy az akkor még elképzelhetetlenül nagy sebességű, 10 Gbit/s-os linkeken is hatékonyan működjön a protokoll. Az IP protokollsaládot folyamatosan bővítették ki, hogy támogassa például a végpontok nagyfokú mobilitását, a veszteséges linken történő hatékony csomagtovábbítást, és a többutas adatátvitelt. Fontos tervezési szempont volt, hogy az új változatokat folyamatosan be lehessen vezetni, azaz az Internet működjön akkor is, ha a hálózati eszközöknek és végpontoknak csak egy része támogatja a bevezetés alatt álló új funkciókat. Ez a fokozatos fejlődés folyamatos működést tett lehetővé úgy, hogy a régebben megvásárolt és üzembehelyezett eszközöket és szoftvereket nem kellett az egyik pillanatról a másikra lecserélni egy új vagy egy jobb hálózati funkció bevezetése miatt. Azonban ez azt is jelenti, hogy a '60-as években megfogalmazott ter-

vezési döntések rányomják a bélyegüket a jelenlegi, egy alapvetően eltérő felhasználási igények által kialakított technológiai környezetre.

A folyamatos fejlődést biztosító evolúciós (evolutionary) tervezési modellel szemben áll a tiszta lappal indító (clean slate) tervezési modell. A clean slate modell a megváltozott környezetre kidolgozandó megoldás során nem veszi figyelembe a korábbi környezetben kidolgozott megoldásokat; ettől remélve egyszerűbb, letisztultabb, és hatékonyabb megoldást. A tartalomközpontú hálózat (CCN) is egy ilyen kísérleti célú, clean slate hálózati protokoll, amely azonban képes IP felett működni. Sőt, a CCN hálózatok elvileg IP csomagokat is képesek továbbítani. Így tehát a CCN egy clean slate hálózat, amelynek azonban az implementálása során ügyeltek a fokozatos bevezethetőségre is.

De mégis mi értelme egy ilyen kísérlet protokoll megismerésének, ha a közeljövőben nem valószínű, hogy valós rendszerekben találkozni fogunk velük? Paradox módon a jelenlegi Internettől teljesen eltérő rendszer megismerése felhívja a figyelmet azokra megoldásokra, amelyeket természetes módon elfogadunk, amelyekre úgy tekintünk, hogy másképpen nem is működhetnének. Például a '60-as években a számítógéphálózatok létrejöttét a rendkívül drága erőforrások megosztása motiválta. Az IP protokoll az erőforrást megosztó és az erőforrást igénybe vevő két fél kommunikációjára épül, emiatt szerepel a forrás és a nyelő IP címe is az IP csomagokban. Az IP cím pedig magát az eszközt azonosítja, ebből következően eléggé macerás a többesadós (multicast) forgalom létrehozása, ahol egy csomagnak több címzettje is van. A '60-as évek után azonban a hálózatok fő funkciója fokozatosan eltolódott az erőforrás-megosztástól a tartalommegosztásig. Régebben az volt az fontos, hogy hol található az adott „nagysebességű” mágnesszalag-olvasó vagy a nagy számítási kapacitással rendelkező szuperszámítógép. Azonban a web böngészése során nem az a fontos, hogy melyik számítógép melyik háttértárában hol található egy adott html fájl, hanem az, hogy mi a fájl tartalma. Vagy kicsit részletesebben kifejtve, hogy a felhasználó böngészője minél hamarabb meg tudja jeleníteni a html oldal, amely mondjuk megbízhatósági okok miatt több helyen is el van tárolva. Multicast, peer-to-peer, vagy tartalom-elosztó (CDN) hálózatokat IP felett is meg lehet valósítani, de ezek a megvalósítások plusz

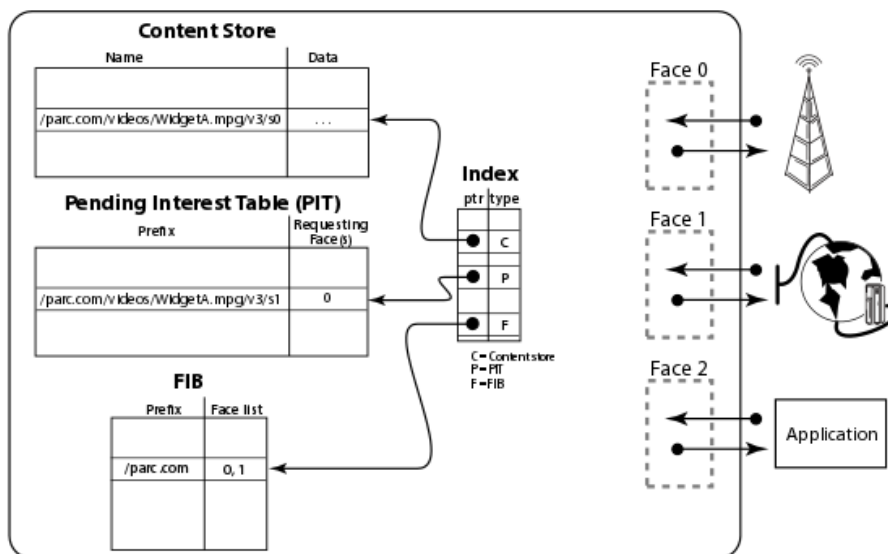


1. ábra. CCN csomagtípusok [1]

hálózati protokollokat, alkalmazás-szintű megoldásokat, vagy plusz hálózati csomópontokat igényelnek. Ugyanezekre a feladatokra a CCN sokkal természetesebb, letisztultabb megoldást nyújt.

## 2. Csomagtípusok

CCN esetén kétféle csomagot különböztetünk meg: adatkérő (interest) csomagokat és magát a tartalmat tartalmazó adatcsomagokat. Ahogy azt az 1. ábra is mutatja, a csomagokban nem a forrás és a cél címei szerepelnek, hanem a tartalmat egyértelműen azonosító neve. Az tartalomnév az IP címhez hasonlóan hierarchikus felépítésű. Az adatkérő csomagokat az adatot igénylő fél küldi a hálózatba. Ezután a hálózat eljuttatja a kérést az adatforráshoz, de a csomag útját megjegyzi az azt továbbító hálózati csomópontok. Az adatforrás egy adatcsomaggal válaszol a kérésre, amit a kéréssel megegyező útvonalon juttat el a hálózat az igénylőhöz. A működés sok részletét még nem tisztáztuk, de már most is látszik két eltérés a TCP/IP-hez képest. (1) Az adatcsomagokra nem érkezik nyugta; az interest csomagra érkezik az adatcsomag, ami felfogható az interest nyugtájaként is. (2) Ellentétben a TCP/IP-vel az odaút és a visszaút ugyanazokon a csomópontokon halad keresztül.



2. ábra. A CCN csomagtovábbítási csomópont vázlata [1]

### 3. A csomagtovábbítási hálózati csomópont

A csomagtovábbításért felelős hálózati csomópont vázlatos felépítését a 2. ábra mutatja be. A csomópont az IP-ben szokásos hálózati interfészek helyett úgynevezett face-eken keresztül kommunikál más csomópontokkal illetve az alkalmazásokkal. Fontos látni, hogy az adatforrásokkal és nyelőkkel ugyanolyan API-n tartja a kapcsolatot a csomópont mint a többi csomóponttal, hiszen az csomagtovábbító csomópont szempontjából az alkalmazás is csak egy kommunikációs fél, ami azonban csak egy bizonyos tartalom-prefixre iratkozik fel.

Ha egy beérkező interest megtalálható a Content Store-ban, azaz a tartalomtárban, akkor a csomópont a tartalomtárban található adattal létrehoz egy adatcsomagot, amit válaszként elküld a kérés irányába és eldobja a kérést. Ha a kérés nem elégíthető ki a tartalomtárból, akkor a kérés bekerül a Pending Interest Table-be (PIT), azaz a függő kérések táblázatába. A táblázatban az található, hogy milyen név prefix-eket honnan kértek. Ezután ha a csomópontban már volt az adott prefix-szel pontosan megegyező bejegyzés, akkor a kérés eldobásra kerül. Ha a kérés nem szerepelt még a PIT-ben, akkor a PIT-be kerül a kérés neve és hogy milyen face-en érkezett a kérés. Ha a kérés

nem szerepelt a PIT-ben, akkor a Forwarding Information Base (FIB, továbbítási táblázat) határozza meg, hogy mi történjen a csomaggal. A FIB-ben a leghosszabb szóeleji egyezést nézve keresi meg a kérés nevét. Találat esetén a prefix-hez tartozó face-ekre küldi ki a kérést a csomópont kihagyva a listából azt a fészt, amin a kérést beérkezett.

Figyeljük meg, hogy az IP hálózatokban beállítási nehézséget okozó várakozási sorok teljesen hiányoznak a csomagtovábbítási csomópontból. A tartalomtárba ugyan bekerülnek adatok, de ezeknek csak későbbi kérések korai kiszolgálásakor van szerepe. A tartalomtár is véges méretű; ha betelik, akkor alapértelmezetten a legrégebben használt elem kerül ki belőle.

Az adattár nagyon hatékony többesadást tesz lehetővé. Többesadáskor az adatnyelők ugyanarra a tartalomra kíváncsiak, ezért ugyanazt a kérést küldik el. Ha a két kérés rövid időn belül követi egymást, akkor a CCN csomópont PIT-jébe két fész kerül a kérés mellé. Ezzel szemben ha az első kérést már kiszolgálta a csomópont, akkor a második kérést (annak megérkezésekor) már a tartalomtárból tudja kiszolgáltatni a csomópont. Mindkét esetben a csomópont, ahova a két kérés beérkezett, csak egyszer továbbítja a kérést az adatforrás felé és a tartalmat csak egyszer küldi el a csomópontnak. Továbbá az adatkommunikációt nem kellett semmilyen többesadós csoport-kiépítési protokollnak megelőznie.

Egy tartalomcsomag mindig egy úton jut vissza a kérőhöz. Azonban egy FIB bejegyzésben több fész is szerepelhet, így a kérés csomag több fészen továbbítódik, ami akkor hasznos, ha például egy kéréshez esetleg több helyen is fellelhető a megfelelő tartalom. (Az emiatt esetlegesen kialakuló körök detektálásában segít a kérés csomag Nonce mezeje. Ezt a mezőt a CCN frissebb változataiból már eltávolították és kördetekciót másképpen oldották meg. Továbbá arra is van lehetőség, hogy bizonyos beállítások mellett a több fész közül csak egyre továbbítsa a csomópont a kérés csomagot.)

A lehallgatás és adathamisítás ellen IP esetén a csatorna titkosításával illetve hitelességének biztosításával lehet védekezni. CCN esetén nem a kommunikációs csatornát védjük hanem magát az adatot. A hierarchikus tartalomnevet különböző mélységben tudják a továbbító csomópontok digitális aláírásukkal ellátni. Sőt, mivel a CCN nyilvános kulcsú titkosítást használ, ezért bármelyik továbbítási csomópont képes a tar-

talom hitelességét ellenőrizni. Emiatt például bizonyos elárasztásos támadásokat elosztottan már a hálózat belsejében ki lehet küszöbölni és az IP-vel ellentétben a támadás detektálása nem hárul kizárólagosan az adatnyelőre. Vegyük például a `/parc.com/george/videos/WidgetA.mpg` tartalomnevet. A videofájl nem fér el egyetlen adatsomagban, ezért azt a küldő szegmentálja. A kérés nevében nem szerepel a szegmens sorszáma, ezért a válasz az első szegmest fogja tartalmazni a következő tartalomnévvel: `/parc.com/george/videos/WidgetA.mpg/s0`. A George felhasználó hitelesítheti az adatsomagot a saját kulcsával, amit pedig `parc` hitelesíthet. A hitelesítési lánc egy elosztott hitelesítési rendszert eredményez központi entitás nélkül. (A video második darabját a `/parc.com/george/videos/WidgetA.mpg/s1` névről lehet lekérni.)

## 4. Többesadás

A fentebb említett működésből látható, hogy CCN technológiával könnyedén megoldható a többesadás problémája. Ha az azonos kérések közül az első már kiszolgálásra került, akkor a több kérést nem kell továbbítani a forrás felé, mivel a tartalomtárból kiszolgálhatók. Viszont, ha két azonos kérés között rövidebb idő telik el, mint az első kérés kiszolgálása, akkor a PIT-be bekerül a második kérés face-e is az adott prefixhez. Ezzel a két megoldással elérhető, hogy a csomópontnak többesadás esetén is csak egyszer kell továbbítani a kérést az adatforrás felé és visszaküldött tartalmat is csak egyszer kell eljuttatni a csomóponthoz. Ezzel jelentősen csökkenthetjük a hálózat terheltségét illetve az azonos kérések kiszolgálása is felgyorsul. Mivel a multicast üzenetküldés a tartalomközpontú hálózatokban nagyon könnyen megoldható illetve rengeteg erőforrás takarítható meg, ezért nagymértékben alkalmas média tartalmak – videó és hang – átvitelére. Például egy videók megosztására alkalmas hálózat kiépítése könnyen elvégezhető a szükséges alkalmazások megléte esetén. Ehhez egy olyan programra van szükség, ami a videó fragmentálása (csomagokra tördelése) mellett a kapott darabokhoz hozzárendeli a megfelelő prefix-eket, amikre majd a felhasználó által használt lejátszó program hivatkozni tud. Egyik ilyen lehetőségk a NDNVideo elnevezésű program, ami kamera képének illetve videófájlok streamelésére is alkalmas CCN hálózaton

keresztül [2].

## 5. Választott CCN komponensek és implementációjuk

Ahhoz, hogy egy ilyen rendszer laborkörülmények között ki tudjunk építeni is és kényelemesen tesztelni tudjuk, szükséges egy olyan program, ami tartalomközpontú hálózat emulálására alkalmas. Erre két lehetőség áll rendelkezésre: a Mini-CCNx<sup>1</sup> és Mini-NDN<sup>2</sup>.

A munka elején a kitűzött cél az volt, hogy egy olyan videómegosztó rendszer állítsunk össze, ami tartalomközpontú hálózaton keresztül működik. Továbbá kíváncsiak voltunk<sup>3</sup> arra, hogy egy ilyen rendszer hogyan viselkedik többesadás és mozgó pontok esetén; illetve, hogyan változik a működés, ha a tartalomtár méretét lecsökkentjük.

Ahhoz, hogy magát a rendszert össze lehessen állítani, szükséges volt a majd használni kívánt programok áttanulmányozása. Meg kellett ismerni a Mini-CCNx, Mini-NDN, CCNx és a NDNVideo programok működését, hogy később könnyedén össze lehessen állítani a tesztelni kívánt hálózatot. A munka során végig Linux Ubuntu 14.04-es operációsrendszert használtunk.

Az eszközök sikeres telepítése után, kezdődhetett magának a rendszernek a mérési rendszernek és a teszhálózatnak az összeállítása. Minden szükséges program szabad hozzáférésű.

AZ NDN hálózat összeállításához a következő programok telepítésére van szükség:

1. Mini-NDN emulációs program,
2. Named Data Networking Forwarding Daemon (NFD és Named Data Link State Routing Protocol (NLSR) eszközök, amik NDN rendszer alapjait szolgáltatják (Content Store, FIB, PIT, stb.)
3. NDNVideo videómegosztó alkalmazás,
4. végül az ezekhez szükséges további Linux eszközök és programok.

---

<sup>1</sup><https://github.com/chestev/mn-ccnx/wiki>

<sup>2</sup><https://github.com/named-data/mini-ndn>

<sup>3</sup>A munka elvégzésében Zelenai Tibor is közreműködött.

A megfelelő működéshez minden programból a megfelelő verzió beszerzésére és telepítésére volt szükség. Ezeket a Linux beépített csomagtelepítőjével illetve a megfelelő programok forráskódból való telepítésével könnyen orvosolni lehetett.

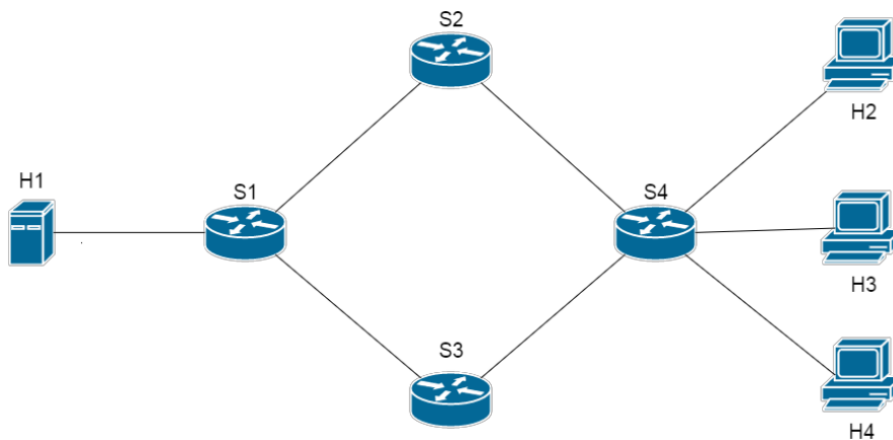
Az NDN rendszer és az NDNVideo tesztelése két számítógép összekapcsolásával oldható meg. Az NDNVideo segítségével a szerver oldalon sikeresen lehetett videót megosztani és a kliens oldalon lejátszani a tartalmat. A Wireshark programmal való vizsgálat viszont megmutatta, hogy az adatátvitel nem az NDN hálózat felett történik illetve eszközökön lévő táblázatok se tartalmaztak bejegyzést a videóra vonatkozóan. A jelenség okát nem sikerült megtalálni.

Mivel az NDN rendszer alatt nem volt sikeres a teszt, ezért egy CCN környezetben is el lett végezve a teszt. Ahhoz, hogy egy CCN rendszert fel lehessen építeni a Mini-CCNx-ben, szükség volt még a CCNx komponensek telepítésére, amik a CCN-hez elengedhetetlen összetevőket tartalmazzák. A CCN rendszer tesztelésekor már a videó megosztása se működött, nem lehetett elérni a streamet a klienssel. A programok alaposabb áttanulmányozása utána kiderült, hogy két hiba miatt nem működött a rendszer:

- A csomópontok továbbítási táblázatába az NDNVideo nem helyezi el a szükséges továbbítási információkat, ezeket egyenként kézzel kell megadni minden csomópont esetén.
- A NDNVideo leírásában a streamelés indításánál megadott `/<név>/video/video` és `/<név>/video/audio` prefixek helyett a `/<név>/streaminfo/video` és `/<név>/streaminfo/audio` prefixeket kell használni, ahol a `<név>` mezőben azt a nevet kell megadni, amivel a streamre hivatkozni lehet.

A szükséges módosítások elvégzése után a CCN rendszer megfelelően üzemelt, a klienssel sikeresen lehetett venni a szerver által biztosított streamet. Ezek után következtek a teljesítménymérések.





3. ábra. A teszhálózat elrendezése

## 6. Teljesítménymérések

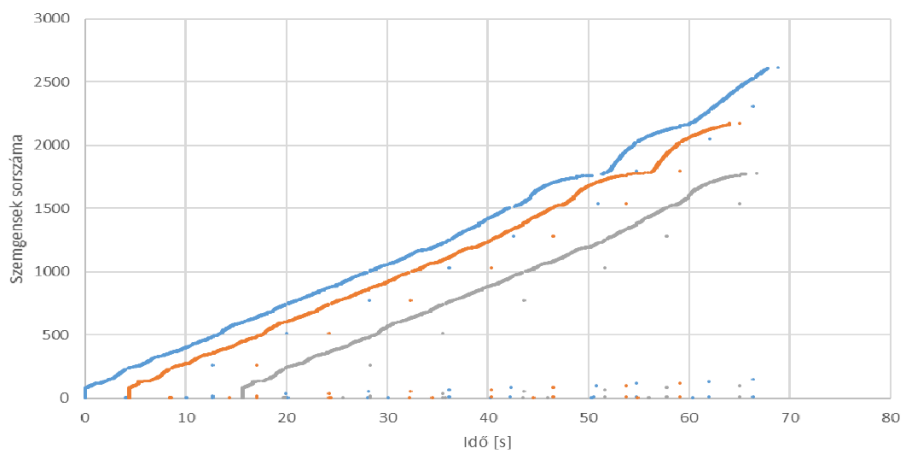
A mérések során használt hálózati topológiát a 3. ábra szemlélteti. A H1 szerveren futó videó streamelése és H2, H3 és H4 hosztokon csatlakoznak ehhez streamhez. A következő méréseket végeztük el a CCN rendszerrel kapcsolatban:

1. a hálózat viselkedésének vizsgálata multicast adatátvitel esetén;
2. hogyan reagál a hálózat a végpont elmozdulására (mobilitás vizsgálata);
3. mi történik, ha Content Store méretet lecsökkentjük és a két lejátszó indítása között megvárjuk, amíg az megtelik.

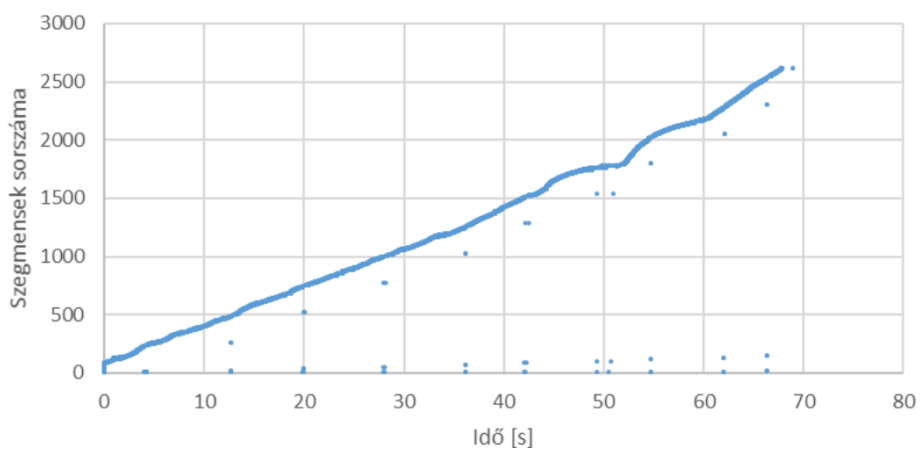
A CCNx tartalmaz egy beépített logolást végző eszközt, ami hálózati eszközönként külön fájllokba rögzíti az eszközökön áthaladó forgalmakat. Az ezekben a fájllokban rögzített prefixekből Excellel való megnyitás és a prefixek megfelelő szétbontása után - mivel a NDNVideo a prefixbe elhelyezi az adott szegmens sorszámát is - már könnyen megkaphatók a szegmensek sorszámai az eredmények kiértékeléséhez.

Az első mérés elvégzéséhez az S2-S4 útvonal ki van kapcsolva, ezzel elérve, hogy csak egy útvonal legyen a H1 és a vevők között. A H1-en elindított streamhez a hosztok egymás után fel lettek csatlakoztatva. A mérési eredmények a következők lettek:

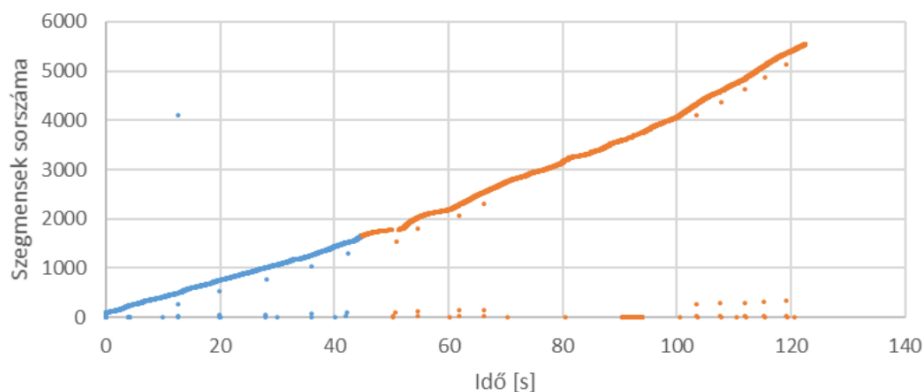
A 4. és az 5. ábra alapján látható, hogy az első kérést az S4 megkapja H2-től, és azt továbbítja is S2 felé. Az útvonalon lévő többi csomópontot nem kell ellenőrizni,



4. ábra. S4 és a lejátszók között átvitt szegmensek



5. ábra. S2 és S4 között átvitt szegmensek



6. ábra. Szegmensek útvonala az útvonal változása esetén

mivel ami S2-S4 útvonal áthalad az át halad az H1-S1 és a S1-S2 útvonalon is. Viszont, amikor H3-tól és H4-től is érkezik kérés ugyanarra a videóra, akkor azokat a kéréseket nem továbbítja S2 felé, hanem a Content Store-jából szolgálja ki. Ennek hatása látható az 1. táblázatban. A H3 és a H4 hosztok kiszolgálása sokkal gyorsabban megtörténik, mint a H2 hoszté. Az ábrákon lévő kiugró értékek nagy valószínűséggel egyéb hibák miatt eldobott csomagok újraküldéséből adódnak.

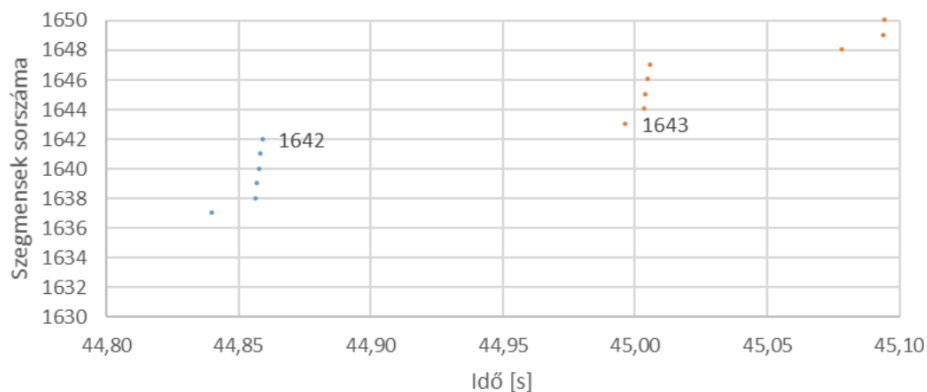
1. táblázat. Az első kérés elküldése és az első adat beérkezése között eltelt idő

Hoszt	H2	H3	H4
Idő [ms]	1,510	0,230	0,270

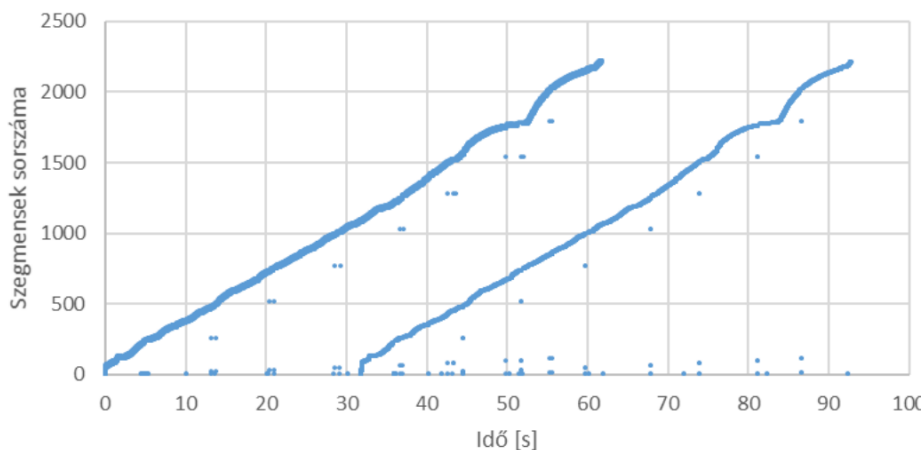
A mérési eredmények alapján is elmondható, hogy tartalomközpontú hálózat alkalmas multicast adatátvitelre, még hozzá nagyon hatékonyan. Látható, hogy H3 és H4 kéréseinek kiszolgálását S4 csomópont el tudja végzeni, így nem kell minden kérésnek végigmenni az egész hálózaton és ezzel jelentős erőforrás takarítható meg.

Második mérés arra vonatkozott, hogy mi történik akkor, ha az egyik végpont elmozdul, vagyis ha módosul az útvonal a szerver és a kliensek között. A mérési elrendezés hasonló az első méréshez, viszont adatátvitel során megváltozik az útvonal S1-S2-S4-ről S1-S3-S4-re. A Mini-CCNx nem támogatja a végpontok mozgatását, de a S2-S4 kapcsolat bontásával és S3-S4 kapcsolat felépítésével ez megoldható.

A 6. és a 7. ábra látható, hogy amikor S2-S4 útvonalat megszüntésakor és S3-S4 útvonalat visszaállításakor, az üzenetek továbbítása is automatikusan átirányítódott az



7. ábra. Szezmensek útvonala az átváltás környékén

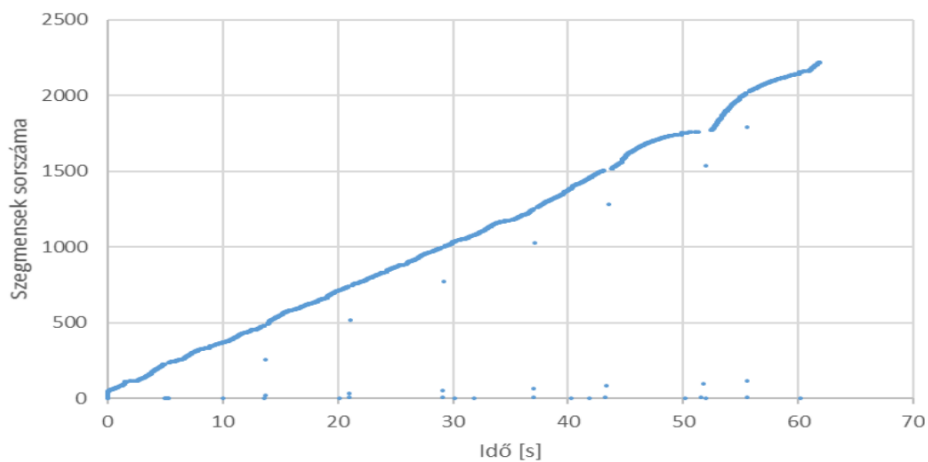


8. ábra. S2 és S4 között az adatátvitel csökkentet Content Store méret esetén

új útvonalra. Ehhez a működéshez viszont szükséges, hogy S4 FIB-jében szerepeljen S2 és S3 felé is vezető továbbítási bejegyzés a videóra vonatkozóan. Például, ha S3 nem szerepel a táblázatban, mint lehetséges útvonal a videótartalom felé, akkor az adattovábbítás megszakad, mivel S4 nem ismer több működő utat H1 felé.

Az eredményekből látható, hogy CCN rendszerek a mobilitást is könnyen tudják kezelni, viszont ehhez elengedhetetlen, hogy az alternatív útvonalak is szerepeljenek az érintett csomópontok FIB-ikben.

Harmadik mérés célja az volt, hogy megnézzük, mi történik akkor, ha a Content Store mérete lecsökken és a két lejátszó úgy indul el, hogy a két indítás között a meg-  
telik a Content Store. A mérés végén a következő eredményt kaptuk:



9. ábra. S1 és S2 között az adatátvitel csökkentet Content Store méret esetén

A mérés során a Content Store mérete le lett csökkentve annyira, hogy csak 500 db bejegyzés férjen el benne. Ezek után H2-ről el lett indítva egy lejátszás, majd nagyjából fél perc múlva H3-ról is el lett indítva egy másik. A 8. ábrán látható, hogy a tartalomtárnak volt ideje megtelni a két indítás között, ezért H3 kéréseit már nem tudta kiszolgálni, ezért a kéréseket továbbítani kellett a következő csomópont felé. Ez jelen esetben az S2, ami már ki tudta szolgálni a kérést, mivel annak a tartalomtárának a mérete nem lett lecsökkentve. A 9. ábrán látható, hogy S2 már csak egyszer küldi el a kéréseket S1 felé.

## 7. Összefoglalás

A kialakított emulációs teszhálózaton elvégzett mérések alátámasztják, hogy a tartalomközpontú hálózatok jó hatásfokkal alkalmazhatóak média információk átvitelére, mivel *(i)* a mobilitásra nem érzékenyek és *(ii)* egyszerűen alkalmazhatóak többesadásra is.

## Hivatkozások

- [1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, „Networking named content,” in *Proceedings of the 5th international*

*conference on Emerging networking experiments and technologies.* ACM, 2009, pp. 1–12.

- [2] D. Kulinski and J. Burke, „Ndnvideo: Random-access live and pre-recorded streaming using ndn,” UCLA, NDN Technical Report NDN-0007, Sep. 2012.