BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai tanszék

Gazdálkodj Okosan! OpenFlow eszközökkel

Készítették:
Czében Attila
Szalay Márk

Konzulensek:
Dr. Heszberger Zalán
Dr. Sonkoly Balázs

Tudományos Diákköri Konferencia
Budapest, 2014

TARTALOMJEGYZÉK

Ábrajegyzék

TÁBLÁZATJEGYZÉK

KIVONAT

Az SDN (Software Defined Networking) koncepció lényege, hogy a gyártók megnyitják hálózati eszközeiket a felhasználók számára, lehetővé téve azok működésének a hagyományosnál sokkal közvetlenebb irányítását. Az új architektúrában az adatsíkot a gyártóspecifikus, egyszerű és gyors adattovábbító hardverelemek jelentik, amelyeket egy harmadik fél által programozható, szoftver alapú központi vezérlő felügyel. Az architektúra kulcseleme a hálózati eszközök és a vezérlő közti kommunikáció egységes megvalósítása, melyre a legtöbb eszközgyártó jelenleg az ONF (Open Networking Foundation) által szabványosított OpenFlow protokollt alkalmazza.

Az SDN hálózati koncepció forradalmian új felhasználási lehetőségeket rejt magában, elterjedése viszont bizonytalan, mivel stratégiai szempontból annak megjelenése nem kedvező az eszközgyártók számára. A versenytársak mögötti lemaradás elkerülése érdekében azonban szinte kivétel nélkül lépést tartanak megvalósításában. Mivel az OpenFlow szabványnak megfelelő új eszközök fejlesztése lassú folyamat, így a gyártók saját, már meglévő architektúráikat próbálják kiegészíteni a szükséges változtatásokkal. Az így kialakuló heterogén környezet miatt nem meglepő tehát, hogy a különböző gyártók eszközei igen eltérő színvonalon képesek csak megvalósítani az OpenFlow szabványban előírt funkciókat. Mindez gondot okozhat az SDN technológiát alkalmazni kívánó ‘early adopter’ hálózatüzemeltetők számára a megfelelő OpenFlow-képes eszközök kiválasztásában.

Tudományos dolgozatunk célja feltérképezni az OpenFlow-képes eszközök valódi OpenFlow támogatottságának mértékét napjainkban, mind az elérhető funkciók mind azok teljesítményének elemzése segítségével. Munkánk kulcselemeként olyan mérési módszertant dolgozunk ki, amely alapján az akár különböző verziójú OpenFlow protokollokat támogató eszközök teljesítménye is összehasonlíthatóvá válik. A kidolgozott módszertant követve méréseket végzünk valós hálózati eszközökön többek között a folyamtáblák kapacitásának, folyambejegyzések mentési idejének és a hálózati portok áteresztőképességének meghatározására. A mérések egyik meglepő eredményeként megmutatjuk, hogy a jelenleg erősen heterogén piaci felhozatal nem mentes az olyan anomáliáktól sem, ahol az akár több nagyságrendben eltérő árú eszközök (akár az eszköz-specifikációknak is ellentmondóan) hasonló teljesítményt nyújthatnak. Következtetéseinkben a napjainkban elérhető OpenFlow-képes hálózati eszközök, felhasználási módok és hálózati elvárások függvényében nem triviális ár-teljesítmény kapcsolatára hívjuk fel a figyelmet.

1. BEVEZETÉS

Az SDN (Software Defined Networking) koncepciónak, azaz a programozható hálózatoknak az egyik elsődleges célja a hálózati eszközök belső működésének (adatsík) és a vezérlő logikának (vezérlősík) a szétválasztása. Ennek előnye, hogy a gyártók továbbra is elrejthetik az eszközeik megvalósításának részleteit, azonban bárki számára lehetőség van a switch-ek és router-ek működésének a hagyományosnál sokkal közvetlenebb irányítására, például új protokollok valós körülmények melletti tesztelésére. Az így létrejött új architektúrában az adatsíkot a gyártóspecifikus, egyszerű, de gyors adattovábbító hardverelemek jelentik, amelyeket egy harmadik fél által programozható, szoftver alapú központi vezérlő (kontroller) felügyel. Az architektúra kulcseleme a hálózati eszközök és a vezérlő közti kommunikáció egységes megvalósítása, melyre a legtöbb eszközgyártó jelenleg az ONF (Open Networking Foundation) által szabványosított OpenFlow protokollt alkalmazza [1].

Az SDN hálózati koncepció forradalmian új felhasználási lehetőségeket rejt magában, fejlődése mégis lassú és bizonytalan. Ennek oka egyik oldalról a nagy eszközgyártók elhúzódó termékfejlesztési ciklusaiban, lassú reakcióidejében keresendő. Ugyanakkor számottevő gátat jelent az új üzleti modell kidolgozatlansága, gazdasági kérdések, ill. a piacra lépés időzítési nehézségei is. További nem elhanyagolható akadályozó tényező persze az is, hogy a valós, ipari célú felhasználási területek egyelőre szűkre szabottak. Nem meglepő tehát, hogy a különböző gyártók eszközei igen eltérő színvonalon képesek csak megvalósítani a szabványban előírt funkciókat. Mindez számottevő gondot okozhat az SDN technológiát alkalmazni kívánó ‘early adopter’ hálózatüzemeltetők számára a megfelelő OpenFlow-képes eszközök kiválasztásában.

Az új koncepció valós alkalmazásainak lassú elterjedésének mélyebb gazdasági és politikai kérdéseivel jelen dolgozatban nem foglalkozunk. Kutatómunkánk a megvalósítások mérnöki problémáira mutat rá, elmagyarázza ennek okait és kitér ezen problémák megoldásának közeljövőben felmerülő kihívásaira is.

Mivel minden gyártó a legkevesebb költség ráfordításával próbálja implementálni a számukra piaci szempontból még nem kedvező OpenFlow protokollt, ezért új hardver megtervezése helyett, a már létező, saját architektúrájukra alkalmazva igyekeznek megvalósítani azt. Az így kialakult heterogén piaci környezetben az egyes eszközök működése és teljesítménye drasztikusan különböző lehet.

A valódi hardveres OpenFlow implementációk ezen problémája égető kérdés, mégis kevés kutatás foglalkozik az eszközök teljesítményének átfogó vizsgálatával.

Talán ezek a vizsgálatok triviálisnak és könnyen kivitelezhetőnek tűnnek, de az új elgondolások számos esetben gyökeresen új, eddig nem használt megoldásokat szültek. Emiatt nem csak újszerű módszerek kidolgozására van szükség, de újabb metrikák bevezetése is célszerű lehet. Mindezt figyelembe véve fontos, hogy az eszközök így kialakuló új tulajdonságai alapján, egy új szempontrendszer szerint összehasonlíthatóak legyenek.

Tudományos dolgozatunk célja tehát, feltérképezni az OpenFlow eszközgyártó oldali támogatottságának mértékét, mind az elérhető funkciók mind azok teljesítményének elemzése segítségével. Munkánk kulcselemeként olyan mérési módszertant dolgozunk ki, mely alapján az akár különböző verziójú OpenFlow protokollokat támogató eszközök teljesítménye is összehasonlíthatóvá válik. A kidolgozott módszertant követve méréseket végzünk valós hálózati eszközökön többek között a folyamtáblák kapacitásának, folyambejegyzések mentési idejének és a hálózati portok áteresztőképességének meghatározására. A mérések egyik meglepő eredményeként megmutatjuk, hogy a jelenleg erősen heterogén piaci felhozatal nem mentes az olyan anomáliáktól sem, ahol az akár több nagyságrendben eltérő árú eszközök (akár az eszköz-specifikációknak is ellentmondóan) hasonló teljesítményt nyújthatnak.

Következtetéseinkben a napjainkban elérhető SDN-képes hálózati eszközök felhasználási módok és hálózati elvárások függvényében nem triviális ár-teljesítmény kapcsolatára hívjuk fel a figyelmet.

A dolgozat következő fejezetében bemutatjuk az SDN koncepciót, kitérünk az OpenFlow protokoll ismertetésére, néhány esettanulmányon keresztül bemutatjuk az alkalmazhatóságát, illetve megemlítjük a megvalósítás nehézségeit. A dolgozat ezután következő fejezeteiben kifejtjük a megalkotott módszertan részleteit, elmagyarázzuk az új megoldások miatt felmerülő kérdéseket, majd a felállított módszerek segítségével elvégezzük a rendelkezésünkre álló eszközök vizsgálatát. A kapott eredményeket összehasonlítjuk és megpróbáljuk azokat értelmezni, illetve a felmerült anomáliákra választ adni. A dolgozat végén összegezzük munkánkat.

2. SDN & OpenFlow

2.1. Software Defined Networking

Az Internet hajnalán alapelvként fogalmazták meg, hogy a rendszer robusztussá tétele miatt ne legyen kitüntetett vezérlőpont a hálózatokban. Az SDN szakít ezzel az elképzeléssel, mivel alapfeltétele a vezérlés és a továbbítás szétválasztása a hálózatok felépítésében. Ennek köszönhetően számos új, izgalmas innovációs lehetőség nyílik meg előttünk. Ha a költséges intelligenciát kivesszük az egyes eszközökből, akkor azok butábbak és olcsóbbak lesznek, viszont továbbra is gyorsan végezhetik a továbbítási alapfeladatukat. Ekkor a bonyolult és "drága" vezérlő logika a hálózat egy vagy több vezérlő elemében kap helyet (kontroller), mely(ek) teljes és pontos képet látnak a hálózati rendszerről.

Ez az új hálózati megközelítés az SDN koncepció, melynek felügyeletével és standardizálásával az ONF foglalkozik. Ezen elképzelés szerint az eddigi sík, egyrétegű szervezést felváltja egy hierarchikus, rétegezett architektúra. A hálózati eszközökben megmarad a gyors adattovábbító hardver, amelyeket centralizáltan egy vagy több vezérlő felügyeli. Az pedig egy jól definiált interfészt kínál magas szintű szoftverkomponensek számára, hogy az eszközök intelligens vezérlését megvalósítsák, és ezáltal elvégezzék a forgalom szervezési, irányítási, menedzsment és monitorozási feladatukat.

A központi vezérlő alkalmazásának előnyei:

- Egy logikailag központosított vezérlő felügyeli az összes folyamszintű döntést, amely egyenként képes konfigurálni a switch-eket, így elkerülhető egy globális policy alkalmazása az összes eszközre
- A központi vezérlő számára látható minden folyam, így képes globálisan optimális hálózati forgalom-menedzsmentre
- A koncepcióban megjelenő kapcsoló eszközök relatíve egyszerűek, mivel az intelligencia a kontroller szoftverekbe kerül, ahelyett hogy az eszközök hardverében vagy firmware-ben lenne megvalósítva

Az SDN architektúra használatának további előnye, hogy a vezérlő centralizáltsága miatt a hálózat adminisztrátora könnyebben felügyelheti, irányíthatja és ellenőrizheti azt. Anélkül, hogy minden egyes eszközzel külön-külön foglalkoznia kellene, elég a vezérlőben elvégeznie a megfelelő beállításokat. Az előzőekben ismertetett okok miatt az adminisztrátor kevésbé költséges és heterogén eszközöket használhat, mégis jóval nagyobb és egyben kifinomultabb kontrollja lesz a hálózata felett. Ezáltal soha nem látott módon lehetőségük lesz skálázható, a folyton megújuló üzleti igényekhez idomuló, jól alkalmazkodó hálózatokat építeni.

2.2. OpenFlow

A legelső és legelterjedtebb SDN szabvány a vezérlősík és az adatsík közti kommunikációra az OpenFlow specifikáció, mely alapvető eleme az új, nyílt szoftveresen definiált hálózati architektúráknak.

Az OpenFlow protokoll feladata a switch-ek/router-ek és a kontroller közti kapcsolat kezelése, ennek köszönhetően a hálózati eszközöket központilag lehet irányítani és menedzselni. Az így létrehozott új architektúrában a switch-ek, mint a fizikai kapcsolatot kiépítő hálózati eszközök és az egy vagy több kontroller, mint a hálózat vezérlője vannak jelen.

Ez a technológia programozhatóvá teszi egy hálózat eszközeit. Az OpenFlow eszközök a hálózatmenedzserek által előre konfigurált szabályok alapján azonosítják a különböző forgalmakat. Ez a megoldás a folyamok segítségével virtuális szeletekre osztja a hálózatot. Ezután a virtualizált hálózat menedzsmentje kiajánlható magas szintű szoftverelemek felügyelete alá.

Összegezve, OpenFlow technológiával programozott hálózatok előnyei:

- a létrejövő absztrakció segítségével kísérleti tesztek végezhetőek működő hálózatokban anélkül, hogy az megzavarná a produkciós forgalmat
- a virtuális hálózati szeletek kiajánlhatóak kutatók számára kutatási célokra
- alacsony az eszközök ára, az OpenFlow switch-ek akár Unix/Linux platformra is telepíthetőek
- az OpenFlow új funkciót adhat a modern Ethernet switch-ek és router-ek TCAM[1]-jének

A legelső verzió, az OpenFlow 1.0.0 kiadása 2009.12.31-re datálódik [2]. A legtöbb eszközgyártó még ezt a legelső verziót implementálta eszközeire. Kutatómunkánk során egy olyan switch-csel is megismerkedünk, ami az OpenFlow 1.3.1 verziót implementálja [3]. Ennek kiadása 2012.09.06-án történt. A jelenleg elérhető legújabb változat 1.4 verziójú (2013.10.15).

Jelen dolgozatnak nem célja az OpenFlow protokoll elemeinek és működésének részletes ismertetése, azonban az alábbiakban összegezzük az alapvető működési tudnivalókat, a további fejezetek könnyebb értelmezéséhez.

[1] Ternary content-addressable memory: Hármas (0, 1 és don't care) tartalommal címezhető memória

Az OpenFlow switch-ek továbbítási döntésekhez a folyamtáblájukban tárolt bejegyzéseket használják. Azt, hogy a bejövő csomagokat merre továbbítsák, a folyambejegyzések határozzák meg, amiket a hálózat vezérlőjétől kapnak meg. Egy folyambejegyzés tartalmaz fejléc[2] (header) mezőket, számlálókat, prioritást, timeout-ot[3] és akciókat. A switch a fejléc mezők alapján keres illeszkedést a bejövő csomagra. Ha van ilyen, akkor a bejegyzés akciójában definiáltakat hajtja végre.

A továbbításra váró csomag vagy pontosan (exact match) vagy wildcard mezőkkel (wildcard match) illeszkedik egy bejegyzésre, illetve az is előfordulhat, hogy nincs folyambejegyzés a beérkező csomagra. Az exact illeszkedés esetén a 1. táblázatban látható összes mező ki van töltve a bejegyzésben, melyek mind megegyeznek a bejövő csomag adataival. Ha a bejegyzésben nincs minden mező kitöltve, akkor azt wildcard bejegyésnek nevezzük. Ekkor a switch a kitöltetlen mezőkre minden beérkező csomagot illeszkedettnek tekint, és csak a konkrét értékkel ellátott mező(k) tartalmát vizsgálja. Az OpenFlow switch exact és a wildcard illeszkedés esetén a bejegyzés akciójában definiáltakat hajtja végre. Amennyiben nem talált egyezést az adott csomagra, akkor azt beágyazza egy OpenFlow csomagba és elküldi a hálózat vezérlőjének (packet-in üzenet), hogy a továbbiakban az döntse el, mit szükséges vele tenni.

1. táblázat: Vizsgálható fejlécmezők

Bejövő port	Ethernet forráscím	Ethernet célcím	Ethernet típus	VLAN ID	VLAN prioritás	IP forráscím	IP célcím	IP protokoll	IP ToS bit	TCP/UDP forráscím	TCP/ UDP célcím

Az OpenFlow 1.3 verziójában (az OF 1.1.0-tól kezdve) már több folyamtábla támogatott, melyekből az OpenFlow switch az illeszkedéseket keresi. Ennek előnye a csomag továbbításának gyorsabb feldolgozása. Ahhoz, hogy ez tényleg jobb lehessen új működésre volt szükség az illeszkedés keresésnél. Minden tábla rendelkezik egy ID-vel. Az OpenFlow switch minden esetben az alapértelmezett (ID0) táblában kezdi az illeszkedés keresést. Találat esetén a végrehajtandó akciókat a switch egy akciógyűjteménybe helyezi el. Ebben a verzióban már szerepelhet a bejegyzések műveletei között egy másik táblára való ugrás utasítás is, melynek következményeként az illeszkedés keresés a megadott táblában folytatódik. Azért, hogy a feldolgozás során ne alakulhassanak ki hurkok a táblák között a `goto` akció csak nagyobb sorszámmal rendelkező folyamtáblára mutathat. Ha az OpenFlow switch végzett az illeszkedés kereséssel, akkor végrehajtja az akciógyűjteménybe elmentett összes műveletet. Az effajta működést pipeline feldolgozásnak nevezzük.

[2] Az 1.0-ás verziójú OpenFlow 12 fejléc mező illeszkedésvizsgálatát írja elő. Ezek segítségével L2 (MAC), L3 (IP) és L4 (TCP/UDP) alapú továbbítást kapunk.

3 Megadott idő letelte után a folyambejegyzés törlésre kerül.

Természetesen még számtalan funkciója van a különböző OpenFlow verzióknak, melyek részletes leírása megtalálható az ONF honlapján [1].

2.3. SDN ESETTANULMÁNYOK

Az SDN koncepció elméleti, illetve az OpenFlow protokoll működésének áttekintése után kitérünk az elképzelés gyakorlati alkalmazhatóságra.

Itt érdemes megemlíteni, hogy esetenként az SDN/OpenFlow szemlélet alkalmazásával megalkotott szolgáltatások, megoldások már korábban is léteztek, az effajta feladatok megoldhatóak voltak. Sok esetben nincs szó arról, hogy drámaian új megoldást kapnánk, mégis jobb, gyorsabb, költséghatékonyabb eszköz kerül a kezünkbe. Ugyanakkor számos esetben olyan feladatokat is meg tudunk oldani, amelyekre eddig eszközeinkkel nem volt lehetőségünk.

A következőkben tehát megvizsgáljuk, hogy különböző szituációkban, miért lehet szükség a koncepció használatára és kitérünk arra is, hogy milyen előnyöket érhetünk el alkalmazása során.

Hálózat virtualizáció – Több bérlős (Multi-Tenant) hálózatok

Adatközpontokban elvárás, hogy dinamikusan hozhassunk létre szegregált hálózatokat egy adott topológiai felett. A hagyományos VLAN alapú megoldás skálázhatósága alacsony – 4096 VLAN osztható ki. Erre a skálázhatósági problémára ad megoldást az OpenFlow. Az új koncepciónak köszönhetően (20-30%-kal) jobb erőforrás kihasználás érhető el [4]. Ezen felül a konfiguráció megváltoztatásának ideje hetekről perces nagyságrendre csökken az automatizálás segítségével.

Hálózat virtualizáció – Kiterjesztett (Stretched) hálózatok

Adatközpontokban felmerülő igény, hogy helyileg távoli pontokat kössünk össze, pl. rack szekrények vagy adatközpontok között, támogatva a VM-ek mobilitását vagy az erőforrások dinamikus újrakiosztását. Ezzel megoldással egyszerűsödhetnek az alkalmazások, nagyobb rugalmasságot nyújtva, nő az erőforrás kihasználtság. A VM-ek transzparens módon mozgathatóak így jól eloszthatóa terhelés. Javítható a helyreállási idő katasztrofális összeomlások után.

Szolgáltatás beillesztés – Szolgáltatás láncolás

Adatközpontokban vagy szolgáltatói WAN hálózatokban merül fel az igény, hogy dinamikusan és automatikusan hozhassunk létre bérlőnkénti Layer 4-7 szolgáltatás láncokat. Például DDoS védelem bekapcsolása támadás esetén, önműködő tűzfal, IPS (behatolás megelőzés) hoszting környezetben, DPI (Deep Packet Inspection) mobil hálózati környezetben. A megoldás segítségével a "provisioning" idő hetekről percekre csökken. A gyors és önkiszolgáló megoldások csökkentik a költségeket és új bevételeket, új szolgáltatásokat eredményeznek.

Monitorozás aggregálás (Tap Aggregation)

Adatközpontok vagy campus-ok sok switch-es hálózataiban nyújt port-szintű átláthatósági vagy hibaelhárítási képességeket anélkül, hogy drága hálózati csomag brókereket (NPB) kellene használni. Ezáltal nagymértékű megtakarítások és kiadás csökkentés érhető el. A kezdeti telepítési költségek alacsonyabbak, nincs extra kábelre szükség a switch-ek és az NPB-k között.

Dinamikus WAN átirányítás

Szolgáltatói vagy nagyvállalati hálózatokban felmerülő igény, hogy nagy mennyiségű bizalmas adatot irányítsunk vizsgáló eszközök felé. Az új koncepció segítségével a már autentikált adatfolyamokhoz kapunk dinamikus hozzáférést a hálózati eszközökbe épített API-kon keresztül. A megoldás segítségével akár 100.000 dollár nagyságrendű megtakarításokat is elérhetünk azáltal, hogy nem lesz szükségünk drága Layer 4-7 tűzfalakra, load-balancerekre.

Dinamikus WAN összeköttetések

Szolgáltatói hálózatok igénye, hogy dinamikusan lehessen létrehozni kapcsolatokat más szolgáltatókkal vagy vállalati ügyfelekkel nagy teljesítményű switch-ek segítségével költséghatékony módon. A kapcsolatok azonnali, önműködő kiépítése csökkenést eredményez a működési költségekben.

Igény szerinti sávszélesség allokálás

A koncepció szolgáltatói hálózatokban lehetővé teszi a szállítási linkeken esetenként felmerülő extra sávszélesség igények dinamikus (on-demand) kielégítését a programozható vezérlés által. A gyors felhasználói önkiszolgálás csökkenti a manuális beavatkozások szükségességét, így csökkennek a működési költségek.

Üzleti és lakossági virtuális edge

Szolgáltatói hozzáférési hálózatokban NFV[4] megoldásokkal vegyítve lecserélhetőek a CPE-k (ügyfél eszközök) egyszerűbb, olcsóbb eszközökre azáltal, hogy az általános funkciók vagy a bonyolult forgalomkezelés átkerül központi POP-okba (szolgáltatás elérési pontokba) vagy adatközpontokba. Ezáltal növekszik az ügyfél eszközök élettartama, növekszik a hibaelhárítási képességünk, csökken a javítások, módosítások, manuális konfigurálások száma, továbbá egyszerűbb új szolgáltatások bevezetése.

2.4. KONCEPCIÓTÓL A MEGVALÓSULÁSIG

2.4.1. ARCHITEKTÚRA

Ahogy az előzőekben kifejtettük az OpenFlow lehetővé teszi a hálózati forgalmak rugalmas vezérlését a tetszőleges folyam-meghatározás segítségével. Szolgáltatói hozzáférési hálózatokban az OpenFlow használható arra is, hogy testre szabható hálózati beállításokat nyújtson VPN-ek létrehozására, IPTV vagy CDN (Content Delivery Network) megvalósítására [5]. Itt és ahogy az esettanulmányokban is látható, gyakran tekintélyes mennyiségű adatforgalom gyors továbbítását kell elvégezni. Ehhez nagy teljesítményű switch-ekre van szükség. Egy OpenFlow switch-nek számos fejlécmező vizsgálatát kell támogatnia, miközben fel kell készülnie a wildcard alapú illeszkedésvizsgálatra is. Ehhez nagyméretű TCAM[5] memóriára és ASIC-ek[6] alkalmazására van szükség. Ezáltal az effajta switch-ek továbbra is nagyon költségesek lennének (a TCAM 80-szor drágább, mint egy azonos kapacitású SRAM [6], ráadásul nagyon energia igényes is [7]).

Ezen költséghatékonysági megfontolások mentén a legtöbb gyártó a saját, meglévő architechtúrájába próbálja beilleszteni az OpenFlow protokollt. Ez a módszer az első, 1.0-ás verziónál, – ami csak a legfontosabb funkciókat határozta meg – még kivitelezhető. A jelenleg használt eszközök erőforrásaival (néhány megkötéssel ugyan, de) képesek vagyunk megoldani a feladatot.

Később az újabb igényeknek megfelelően fejlesztették a protokollt, újabb OpenFlow szabványok jelentek meg, melyek már több funkció (ezen belül például még több fejlécmező illeszkedésvizsgálatának) támogatását írták elő. Ennek a kibővített funkció-készletnek az implementálása már nehézkes a rendelkezésre álló hardverelemekkel. Néhány funkció esetén ez lehetetlen vagy csak nagy kompromisszumok árán lehetséges, mivel a megvalósítások nem

[4] Network Functions Virtualization: Virtializáció révén egyes hálózati funkciók csoportjaiból hoz létre építőelemeket, melyek összekötésével vagy egymás után láncolásával hálózati szolgáltatások alakíthatóak ki.
[5] A TCAM egy speciális memória, ami wildcard illeszkedésvizsgálatot tesz lehetővé – a bejegyzések számától függetlenül – 1 óraciklus alatt
[6] Application-specific integrated circuits: alkalmazás-specifikus integrált áramkör

illeszkednek a meglévő, hagyományos eszközök hardver környezetébe. Emiatt létezik néhány javaslat arra, hogy az alapoktól kellene újratervezni a koncepciónak megfelelő hardver architektúrákat [8].

Ezeket a problémákat megkerülve, tehát érdemes lehet új eszközök tervezése, fejlesztése, de sajnos ez nagyon költséges és lassú folyamat, így nem biztos, hogy megéri az eszközgyártóknak. Ennek eldöntése bonyolult kérdés, aminek gyakran nem csak mérnöki vagy pénzügyi, de sokszor politikai, gazdaságpolitikai vagy egyéni döntésen alapuló aspektusai is vannak. Jelenleg kérdéses tehát, hogy a piaci szereplők el akarnak-e indulni ebbe az irányba.

2.4.2. CORE-EDGE SZEPARÁCIÓ

Azonban, ha mélyebben megvizsgáljuk a kérdést és megpróbáljuk particionálni azt, a probléma részfeladatokra bontásával jobban kezelhetővé válik. Ugyanis egy hálózat belsejében nem elvárás a folyamok magas szintű menedzselése. Nem szükséges – az új szabványban meghatározott nagyszámú fejlécmező illeszkedésvizsgálat alapú – kifinomult döntések meghozására, csak gyors kapcsolásra van szükség. Ezért a hálózat belsejében (core) elég lehet az OpenFlow 1.0 szabványban előírt egyszerű működés jelenléte, amelyet viszonylag kompromisszum mentesen képesek vagyunk megvalósítani.

A kibővített, újabb verziókban előírt funkciókat elég a hálózat peremén (edge) alkalmazni. Itt tehát van jelentősége az 1.3, 1.4 verziójú OpenFlow szabványoknak is. Ez az a pont, ahol az aggregálás után összegyűlnek a folyamok. Itt azonban rendelkezésre áll a gyors, párhuzamosítható, szoftveres, a protokoll minden funkcióját támogató, dinamikus feldolgozás lehetősége. Így nem feltétlenül szükséges ezeket az újabb szabványokat hardveres alapokra implementálni. A szoftveres megoldások viszont tipikusan könnyen elhelyezésre kerülhethetnek PoP-okban (Point of Presence), adatközpontokban.

2.4.3. IMPLEMENTÁCIÓS LEHETŐSÉGEK

Az előzőek alapján már jól látható, hogy egy OpenFlow eszköz teljesítménye nagymértékben függ a rendszer architektúrájától. Általában 3 népszerű módon szokták implementálni az OpenFlow-t:

- általános célú CPU + DRAM
- FPGA[7] + SRAM
- ASIC + TCAM

[7] A felhasználás helyén programozható logikai kapumátrix

Minden architektúrának megvannak a maga előnyei és hátrányai is. A következőekben megpróbáljuk bemutatni a különbségeket.

Általános célú CPU DRAM-mal:

Ezt alkalmazzák a legtöbb x86-alapú implementációnál, mint például hypervisor-ban[8] futó virtuális switch-ek esetén is. Ennél a rendszernél minden adatsíkban lévő funkciót az általános célú CPU végez el, az tehát teljesen szoftveresen kezelt.

A folyamtábla általában DRAM-ban tárolt, ami jól skálázható és igen olcsó. A rendszer nagy előnye, hogy folyambejegyzések millióit képes tárolni és kezelni. Ugyanakkor, annak ellenére, hogy különböző szoftveres algoritmusokkal az illeszkedésvizsgálat sebessége optimalizálható, az elérhető átviteli kapacitás a folyambejegyzések növekedésével romlik. Az architektúra hátránya tehát, hogy az átviteli sebesség meglehetősen limitált, több Gigabit Ethernet (továbbiakban GbE) port vonali sebességű kiszolgálása problémás.

FPGA SRAM-mal:

Ez a rendszer általában két részből áll. Egy komponens kezeli az adatsíkbeli logikát, ami a forgalmat továbbítja. A másik komponens a vezérlő CPU, ami a menedzsmentet és a protokollokat kezeli. Ezek a komponensek két különálló chipben kerülnek megvalósításra.

Az FPGA alapú rendszerek nagysebességű csomagvizsgálatot és továbbítást nyújtanak, ehhez azonban nagy sebességű memória-hozzáférésre van szükségük. A megvalósítások így gyors SRAM-ot használnak az adatsíkban, ami jóval drágább, mint a DRAM. Ráadásul jóval kevesebb bejegyzést tudnak tárolni – a tipikus érték ezres nagyságrendű. Ez a megoldás ugyanakkor már képes több GbE port vonali sebességű kiszolgálására is. Viszont azt is meg kell említeni, hogy a legjobb eredmény eléréséért az FPGA-t érdemes testre szabni a felhasználástól függő módon.

ASIC TCAM-mel:

Amikor nagyszámú GbE vagy akár 10GbE portot akarunk duplex módon kapcsolni a szükséges backplane kapacitás könnyen elérhet olyan mértéket, ahol 176Gbps-os (48xGbE és 4x10GbE port) vagy akár 1,28Tbps-os (48x10GbE és 4x40GbE port) memória sebességre van szükség. Ezt a sebességet a tipikus DRAM és SRAM-ok nem képesek tartani.

[8] A hypervisor olyan szoftver vagy hardver, ami virtuális számítógépek futtatását végzi.

A TCAM megfelelően gyors erre a feladatra, viszont alkalmazásának sok hátránya van. Egyrészt rendkívül drága, másrészt nagy helyet foglal, ráadásul sok energiát is fogyaszt. Emiatt a gyártók eszközeikben a lehető legkevesebbet helyezik el belőle. Így tipikusan néhány száz, esetleg 1-2 ezer folyambejegyzés menthető el ilyen módon.

Hibrid megvalósítások:

Sok esetben a fentebb részletezett megoldások együtt jelennek meg egy termékben. Lehetőség van például a TCAM kapacitás határáig folyamok gyors, vonali sebességű kapcsolására. Amennyiben nagyobb számú folyamot kezelnénk, azt SRAM alapú szoftveres feldolgozással végezhetjük el. Ilyen a HP 3500 yl switch is, amit a tesztek során tapasztalni is fogunk.

A másik későbbiekben tesztelt hálózati eszközeink a MikroTik 750GL és RB2011, melyek MIPS-BE architektúrán alapulnak. Mindkettő Atheros8327 switch chippel és 64MB SDRAM-mal rendelkezik. A különbség a processzorban van. A 750GL egy 400 MHz-es (AR7242), míg az RB2011 egy 600MHz-es (Atheros 74K MIPS) CPU-val rendelkezik. Mindkettő eszközön a Linux v3.3.5 kernelen alapuló RouterOS rendszer fut. Ezek a switch-ek az általunk felsoroltak közül a legelső (CPU + RAM) módon implementálják az OpenFlow-t, azonban a gyártó belső megoldásai, vagyis a pontos algoritmusok a csomagok feldolgozására és továbbítására nem ismertek.

3. MÓDSZERTAN

3.1. BEVEZETŐ

A mai szoftver vezérelt hálózatok világában OpenFlow nélkül nem létezik SDN. Ez a protokoll a kontroller (vezérlősík) és a hálózati eszközök (adatsík) közötti kommunikáció ellátására szolgál, ezért kijelenthető, hogy a koncepció egy kulcsfontosságú elemét valósítja meg. Ha hálózati eszközök SDN megvalósításra való alkalmasságát vizsgáljuk, akkor ma lényegében az OpenFlow protokoll implementálását kell tesztelünk az adott terméken.

Napjaink két legnagyobb problémája – és egyben legnagyobb kihívása is az SDN effajta megközelítésének – a jelenleg elérhető kapcsoló hardverekben és magában az OpenFlow protokollban keresendő. Egyrészt a manapság használt hardver switch-ek elég rugalmatlanok, mivel az OpenFlow protokoll feldolgozási megoldását (Match-Action) nem támogatják minden fejlécmező esetén. Másrészt az OpenFlow specifikáció – a folyamatos fejlesztés ellenére, – egyelőre csak egy részét támogatja a lehetséges csomagfeldolgozási műveleteknek.

Az SDN által nyújtott előnyök között van a gyártófüggetlenség, mivel az elkülönült vezérlősík (kontroller) elfedi az adatsík különbségeit és homogén felületet nyújt a magasabb absztrakciós szinten lévő szoftverelemek számára. Amíg ez az elképzelés elméletben működőképesnek bizonyul, addig az eszközgyártók a valódi hardveres megvalósításaikban még nem érték el ezt a fejlettségi szintet. Mivel minden gyártó a legkevesebb költség ráfordításával próbálja implementálni a számukra piaci szempontból még nem létfontosságú OpenFlow protokollt, ezért az új hardver megtervezése helyett, a már létező, saját architektúrájukra alkalmazva igyekeznek megvalósítani azt. Az így kialakult heterogén piaci környezetben az egyes eszközök működése és teljesítménye drasztikusan különböző lehet.

A valódi hardveres OpenFlow implementációk ezen problémája égető kérdés, mégis kevés kutatás foglalkozik az eszközök teljesítményének átfogó vizsgálatával. Tudományos munkánk célja az így kialakult űr kitöltése, egy olyan módszertan megalkotásával, mellyel az eszközök OpenFlow működésének és ennek következtében magának az eszközök teljesítményének értékelése is megvalósulhat. Az elért eredmények ugyanakkor a különböző termékek összehasonlíthatóságát is lehetővé teszik, ezáltal megakadályozhatóvá válik, hogy egy rosszul kiválasztott switch vagy router negatívan befolyásolhassa egy SDN alapú rendszer működését.

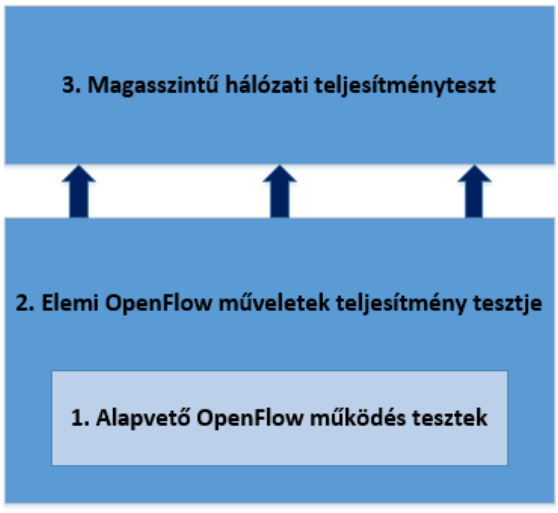
Itt jegyezzük meg, hogy az ONF-nek létezik egy OpenFlow megfelelőségi programja, mellyel az eszközgyártóknak ad lehetőséget arra, hogy demonstrálhassák eszközeik szabványnak megfelelő működését. Ezek a tesztek ugyanakkor csak a szabványban előírt

fejlécmezők illeszkedés-vizsgálatának implementálását ellenőrzik. A mi munkánk azonban ennél sokkal többről szól. Méréseinkben elkerüljük a formális, validáció-szerű tesztelési módszereket. Nem feladatunk az efféle konformancia vizsgálat, hiszen a funkciók puszta megléte nem mond semmit az eszközök különböző teljesítménybeli paramétereiről. Hiába kap egy eszköz megfelelőségi logót, ha ez nem ad választ arra, hogy valós környezetben, valós feladatok elvégzésekor milyen teljesítménnyel képes működni. Ezért is van szükség egy új módszertan megalkotására, amely segítségével választ kaphatunk a felmerülő kérdésekre.

3.2. MÓDSZERTAN SZERKEZETE: MÉRÉSEK CSOPORTOSÍTÁSA

A módszertan részeit alkotó méréseket három fő csoportba osztottuk be. Úgy gondoljuk, hogy egy SDN hálózatépítő számára ezt a három kategóriát érdemes alaposan megvizsgálni az eszköz OpenFlow támogatottságának pontos ismeretéhez. Az általunk definiált három kategória célja, a különböző típusú tesztek szoros összefüggőségeinek feltárása. Ezek a mérések egyrészt a fontos funkciók megfelelő implementálását értékelik, másrészt olyan – sok esetben saját magunk által definiált – metrikákat mérnek, amelyek alapján különböző OpenFlow-képes eszközök összehasonlíthatóvá válnak. E három logikai csoport a következő:

- **Alapvető OpenFlow működési tesztek**
- **Elemi OpenFlow műveletek teljesítmény tesztjei**
- **Magasszintű hálózati teljesítmény tesztek**



1. ábra: Mérések csoportosítása

Az OpenFlow belső működésével az első kettő csoport foglalkozik. Annak, hogy ezeket a teszteket kettő külön részre bontottuk fel fontos oka van. Az *Alapvető működési tesztek* azokat a szükséges előismereteket fogalmazzák meg, amik elengedhetetlenek egy olyan hálózati vezérlő szoftver programozásához, amely az adott eszközzel valósítja meg az

SDN-t. Továbbá az *Elemi OpenFlow műveletek teljesítmény tesztjei* azokat az értékeket tárják fel a tesztelő számára, melyek eredményei meghatározzák az OpenFlow belső működését és ebből következően a switch különböző feldolgozási sebességeit is. Végül az utolsó, *Magasszintű teljesímény tesztek* alatt azokat a méréseket éretjük, melyek átfogó képet adnak a switch működéséről és teljesítményéről az OpenFlow szakértelemmel nem rendelkező vásárló számára is.

3.2.1. ALAPVETŐ OPENFLOW MŰKÖDÉSI TESZTEK

Az SDN egy új réteget vezet be a hálózati infrastruktúra fölé, ami segítségével elfedhetjük a hardverek különbségeit, így egységes felületen (interfészen) keresztül érhetjük el annak szolgáltatásait.

Annak ellenére, hogy az SDN a gyártófüggetlenség, vagyis a vezérlési síkon és a magasabb absztrakciós szinten működő szoftverelemek hordozhatóságának megvalósulásával kecsegtet, a tapasztalataink szerint ez a gyakorlatban mégsem valósul meg igazán.

Habár az OpenFlow szabvány jól definiálja a működést (a protokoll üzenetváltásokat), a kötelezően támogatandó funkciók mellett számos egyéb használata csak opcionális (vagy csak ajánlott). Emiatt a tervezők számára, egy OpenFlow hálózat vezérlőrendszerének elkészítéséhez szükségük van az adatsík ismeretére is, azaz az eszközök működésének előzetes megismerésére. Az ilyen ajánlott funkciók közül gyűjtöttük össze a legfontosabbakat és legkritikusabbakat a munkánk során. Ezek pontos ismerete nélkül a felépített rendszer hibás működést produkálhat.

E gondolatok mentén a következő négy tesztet végeztük el ebben a csoportban:

- **Switch által támogatott táblák száma**
- **Egy folyam bejegyzés mentésének ellenőrzése**
- **Teli tábla jelzésének ellenőrzése**
- **Eszközök hardveres és szoftveres feldolgozás-kritériumainak ellenőrzése**

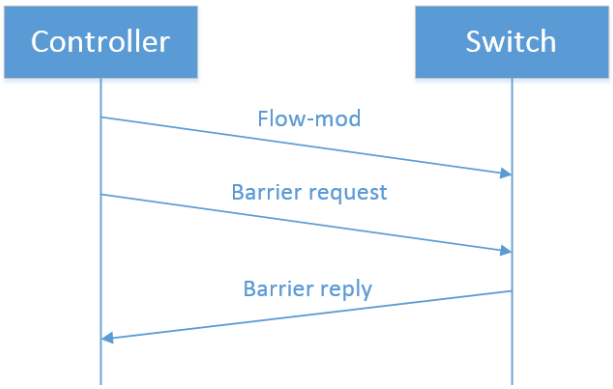
Switch által támogatott táblák száma

Fontos annak ismerete, hogy az adott switch mennyi folyamtáblával tud együtt dolgozni. Az eredeti 1.0-ás verziójú OpenFlow csak egyet támogatott, ez azonban a tábla túl nagy mérete esetén teljesítménycsökkenést okozhat, hiszen egy teli táblánál az aktuális folyambejegyzés megtalálásának ideje hatással van a csomag továbbításához szükséges időtartamra. Ennek elkerülése végett az újabb verziókban ez a funkció módosításra került. Az OpenFlow 1.3-as protokoll már több folyamtáblával dolgozik együtt, így a keresési folyamat

felgyorsulhat, míg a több tábla logikailag továbbra is egy marad. Ez a gyorsulás a pipeline feldolgozás működésének köszönhető. Ennek részleteit már kifejtettük az előző fejezetben. A táblák számának ismerete tehát nagymértékben befolyásolhatja a hálózat megtervezését, a minél gyorsabb működés eléréséért.

Egy folyam bejegyzés mentésének ellenőrzése

Az OpenFlow switch működéséhez elengedhetetlen, hogy a folyambejegyzések korrekt módon (hibátlanul) elmentésre kerüljenek, mivel ez a csomagtovábbítás alapja. Reaktív működés esetén folyambejegyzés új igény felmerülésekor generálódik a kontrollerben. A kontroller egy flow-mod üzenetben elküldi ezt a switch-nek, amely a bejegyzést saját implementációjától függően, valamilyen módszer szerint elmenti a folyamtáblájába. Szabvány szerint a mentés sikerességéről – ha azt a kontroller Barrier Request üzenetben kérte – Barrier Reply üzenetben válaszol. Méréseink során ezt az üzenetváltást vizsgáltuk, melyet a 2. ábra szemléltet.



2. ábra: Bejegyzés mentési üzenetváltások

Lépések:

1. Flow-mod üzenettel küldtünk folyambejegyzést a switch számára
2. Barrier Request üzenettel utasítjuk folyamatban lévő processzei feldolgozására
3. Barrier Reply üzenetben nyugtázza a bejegyzés feldolgozásának sikerességét

Teli tábla jelzése ellenőrzése

Az OpenFlow szabvány számos eleme csak ajánlott, nem pedig kötelező. Véleményünk szerint az ebbe a kategóriába tartozó elemek közül, az egyik legfontosabb a teli tábla jelzés küldésének ellenőrzése. Ennek ismerete nélkül a kontroller rossz programozása hibás működést eredményezhet, így létfontosságú ezen adat tervezés előtti ismerete.

A probléma vizsgálatának módszere a következő: Addig küldünk újabb és újabb flow-mod üzenetet – bennük az elhelyezni kívánt bejegyzésekkel – a switch-nek, amíg az lehetséges. Ha a táblák megteltéről az eszköz hibaüzenettel válaszol a kontroller számára, akkor az adott eszközben implementálták a teli tábla jelzés ajánlást. Fontos megjegyzés a módszer alkalmazásakor, hogy létfontosságú a bejegyzések timeout mezőinek megfelelő beállítása a bejegyzések automatikus törlődésének elkerülése végett.

Eszközök hardveres és szoftveres feldolgozás-kritériumainak ellenőrzése

A gyártók költséghatékonyság, illetve a termékfejlesztési ciklusok lassúsága miatt jelenleg a meglévő eszközeik architektúrális alapjaira tervezik az OpenFlow implementációikat. Ideális esetben a protokoll megoldásai implementálhatóak az eddigi programozható hardverelemek segítségével is. Mivel azonban ezek tervezésekor a hagyományos továbbítási műveletek gyors feldolgozása volt a cél, ezek OpenFlow esetén nem mindig használhatóak. Ilyen esetekben a műveletek elvégzését az eszköz menedzsment processzorára kell bízni. Más szóval, ezek a csomagok nem hardveresen, hanem szoftveresen kerülnek feldolgozásra. Az eszközök menedzsment CPU-ja azonban alacsony teljesítményű, így az elérhető átviteli sebességek nem érik el a vonali sebességet, akár egy nagyságrenddel is kisebbek lehetnek.

Másrészről, ellentétben a hagyományos MAC cím illetve IP cím alapú megoldásokkal, az OpenFlow szabályok nem célonként, hanem folyamonként meghatározottak. Egy tanulmány [9] szerint például a tradicionális Ethernet kereséshez képest 1 nagyságrenddel nagyobb számú (10:1 arány) szabályt szükséges tárolni. A wildcard bejegyzések használata ugyan csökkenti a bejegyzések számát, de az ilyen jellegű, általánosabb szabályok a folyamtovábbításra vonatkozó befolyásunkat is redukálják [10]. Másrészt, míg az exact match keresések implementálhatóak gyors hash táblás módszerekkel, addig a wildcard szabályok gyors feldolgozásához TCAM memóriát kell használunk. Mivel ez a fajta memória drága erőforrás, ezért a jelenleg használt eszközökben kevés áll rendelkezésre. Így ez a lehetőség a szükségesnél nagyságrendekkel kevesebb szabály tárolására alkalmas. Azokban az esetekben, amikor ezeket a gyors hardveres megoldásokat nem tudjuk használni, szoftveres algoritmusokat kell alkalmaznunk a csomagok továbbítása során. Ugyanakkor, mint ahogy azt már az előzőekben leírtuk, ezek a megoldások alacsony teljesítményűek. A tesztben az eszközök ezzel kapcsolatos hiányosságait, megkötéseit tárjuk fel.

3.2.2. Elemi OpenFlow műveletek teljesítmény tesztelése

A különböző gyártók által készített OpenFlow termékek, mind-mind különböző teljesítménnyel képesek ellátni a belső működéséhez szükséges funkciókat. Ez tehát a rendelkezésre álló erőforrások mérését teszi szükségessé. Míg az ipari szereplőket ezek a

mérések kevésbé érdeklik, mégis célszerű ezeknek az elemi műveleteknek a feltérképezése, mivel a switch magasabb szintű működése (pl: áteresztőképesség), erősen függ az OpenFlow belső működésétől.

Véleményünk szerint a következő három teszt elvégezése elengedhetetlen ebben a csoportban:

- **Táblakapacitás teszt**
- **Flow-mod teljesítmény teszt**
- **Packet-in teljesítmény teszt**

Táblakapacitás teszt

A táblakapacitás teszt alatt a switch által támogatott táblák méretére vagyunk kíváncsiak, vagyis, hogy mennyi a maximálisan elmenthető folyambejegyzés száma.

Hálózatok, és általánosságban rendszerek tervezésekor fontos kérdés, hogy a megoldás mennyire skálázható. Ennek egyik eleme a tároló kapacitás skálázhatóságának mértéke. A felhasználhatóság szempontjából kritikus, hogy nagyszámú szabály (folyambejegyzés) tárolására van-e megfelelő méretű memóriakapacitása az eszköznek.

Ez a kérdés erősen függ az éppen aktuális OpenFlow verziótól és a hardveres megvalósítástól egyaránt. Amíg az OpenFlow 1.0 protokoll csak egy folyamtáblát támogat, addig az ettől eltérő verziók már akár többel is együtt dolgozhatnak, aminek célja a keresés gyorsítása úgy, hogy minden folyamtáblában különböző típusú folyambejegyzések tárolódnak.

Mindezek alapján tehát szükségünk van minden táblatípus és több fajta folyambejegyzés esetén elérhető mentésszám tesztelésére.

Flow-mod teljesítmény teszt

Itt a különböző flow-mod üzenetek végrehajtásának idejére vagyunk kíváncsiak. A flow-mod egy kifejezés azokra az OpenFlow üzenetekre, melyeket a kontroller küld a switch számára, hogy hozzáadjon, módosítson, vagy töröljön bizonyos bejegyzéseket a switch folyamtáblájából.

Ahogy a hálózatok növekednek, egyre több forgalmi folyam-változást kell kezelniük. Beszámolók szerint, egy 1500 szervert tartalmazó adatközponti klaszterben másodpercenként átlagosan 100.000 folyam érkezik be [11]. Más kutatás rámutat, hogy szerver rack-enként másodpercenként 10.000 folyam beérkezésére kell felkészülni [12].

Felmerülhet az igény, hogy minden (vagy néhány hasonló) kéréssel egyénileg, mikro-folyam szinten szeretnék foglalkozni. Ez maga után vonja, hogy dinamikusan és gyorsan nagyon sok új folyambejegyzést szükséges installálni.

Ilyenfajta használat esetén a hagyományos hálózati switch-ek korlátozott számítási kapacitása illetve saját adat és vezérlési síkjuk közötti kommunikáció véges sebessége is problémát okoz. Ez tehát limitálja az új folyambejegyzések feldolgozásának sebességét. Szolgáltatások skálázhatósága ebben az esetben attól függ, hogy az eszköz hány bejegyzést képes kezelni és milyen gyorsan képes elmenteni azokat.

Ez a teszt a kontroller által küldött flow-mod üzenet (hozzáadó, módosító vagy törlő bejegyzés) és ennek a bejegyzésnek a használata között eltelt idő mérésével foglalkozik. Az így megkapott időből kiszámítható egy, az OpenFlow belső működését meghatározó paraméter, az egy másodperc alatt feldolgozott flow-mod üzenetek száma, aminek mértékegysége a **flow-mod/másodperc.**

Packet-in teljesítmény teszt

A protokoll működésének jellegéből adódóan azokban az esetekben, amikor a switch nem talál folyamtáblájában illeszkedő bejegyzést egy beérkező csomagra, akkor az adatsík az eszköz vezérlősíkját kéri a csomag fejlécmezők becsomagolására, majd egy packet-in üzenetben a kontrollernek történő elküldésére. Ezek az üzenetek alapjai számos fontos alkalmazás működésének, mint például a hálózat felderítés és a MAC cím tanulás.

A Packet-in teljesítményének szűk keresztmetszete általában az eszközön belül található, erősen függ a CPU feldolgozási sebességétől, vagy egyéb erre a célra implementált hardver működésétől. Költséghatékonyság miatt a legtöbb switch-ben a menedzsment CPU relatíve lassú, mert nem ilyen feladatok ellátására tervezték. Másrészről egy hagyományos switch processzora és az adatsíkja közötti csatorna ritkán használt, így nem tervezik nagy kapacitásúra. Mivel ez gyenge pontja az OpenFlow protokoll működésének, elengedhetetlenné vált a mérése. Az eszközök összehasonlíthatóságának érdekében ez esetben a **packet-in/másodperc** mértékegység definiálása vált szükségessé.

3.2.3. MAGASSZINTŰ HÁLÓZATI TELJESÍTMÉNY TESZTEK

A magasszintű teljesítmény tesztek alatt azokat a mérések értjük melyek átfogó képet adnak a switch működéséről és sebességéről. Ezek többnyire forgalom tesztek különböző átviteli módok és terhelések mellett. Az így megkapott eredmények értelmezéséhez nem szükséges OpenFlow szakértelem, mivel az “egyszerű” termékvásárló számára az elsődleges tulajdonságokat jelenítik meg. A magasszintű kifejezés utal rá, hogy ezek a tesztek komplex

(összetett) műveletek, eredményük erősen függ az *Alapvető* és az *Elemi OpenFlow műveletek* működésétől, mégis a végeredményben ezek a részletek nem jelennek meg.

Az alábbi tesztek tartoznak ebbe a kategóriába:

- **RTT mérés**
- **Áteresztőképesség mérése 1 folyam esetén**
- **Áteresztőképesség mérése terhelt környezetben**
- **Átkapcsolási teszt**
- **Multicast / Broadcast teljesítmény**

RTT mérés

Az RTT (Round-Trip Time) mérése értelemszerűen a switch-re bekötött eszközök közötti oda-vissza út idejét számolja az ICMP protokoll segítségével. Ezzel általános képet kapunk az eszköz reakcióidejéről.

Ahogyan az előzőekben ismertetésre került, az OpenFlow esetében szétválik a vezérlési logika az adatsíktól. Ezzel számos előnyre teszünk szert és nagy flexibilitást érünk el, de felvetődik a kérdés, hogy vajon hatással van-e ez a teljesítményre? Túl nagy késleltetést ad-e az OpenFlow feldolgozás vagy az overhead csak minimális és vállalható a negatívum? Ezekre a kérdésekre kapjuk meg a választ a mért eredmény és az eredeti (OpenFlow nélküli switch mód) RTT idejének összehasonlításával.

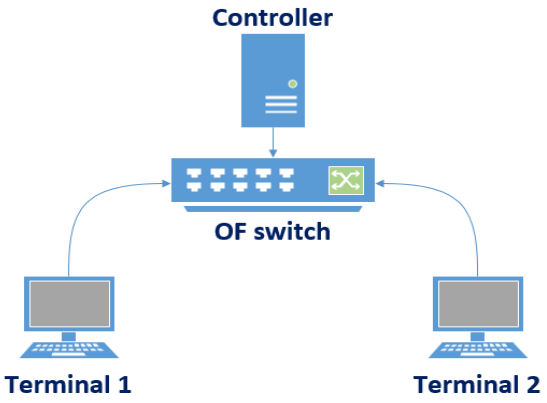
Áteresztőképesség mérése 1 folyam esetén

Egy vásárló számára az eszköz csomagtovábbítási sebessége a legérthetőbb és legfontosabb paraméter, ezért elengedhetetlen az OpenFlow feldolgozás sebességének mérése TCP és UDP forgalom esetén. Fontos tudnivaló, hogy az áteresztőképesség erősen függ a switch aktuális állapotától és beállításaitól, ezért érdekes különbségek adódhatnak az eredményekben különböző folyamtábla feltöltöttségnél, vagy eltérő (wildcard és exact match) bejegyzések esetén. Mindezek miatt ez a teszt az OpenFlow switch állapotától függően további altesztekre bontható.

Az alapvető switch állapotok a következőek lehetnek:

- (majdnem) üres folyamtábla
- teli folyamtábla
- wildcard bejegyzések
- exact match bejegyzések
- szoftveres feldolgozású bejegyzések
- hardveres feldolgozású bejegyzések

Egy lehetséges mérési elrendezés a 3. ábrán látható:

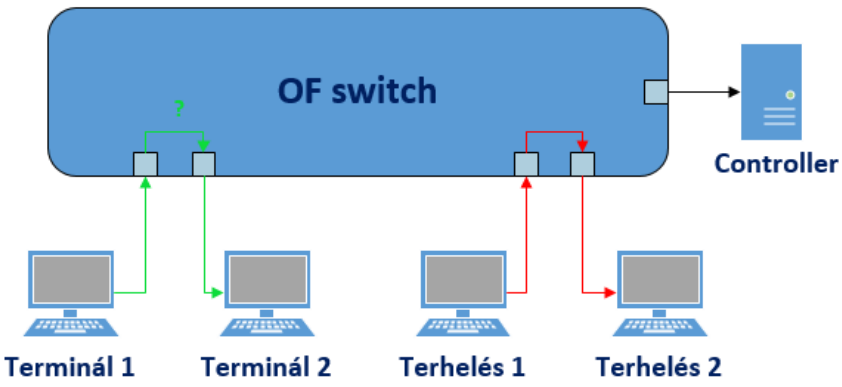


3. ábra: Áteresztőképesség topológia

Áteresztőképesség mérése terhelt környezetben

Az előző pontban kapott eredmények ideális esetben valósak. Ebben a helyzetben az ideális eset jelentése, az egyszerre csak egy folyam átvitele az adott switch-en keresztül. Azonban ez a feltétel a hétköznapi működés során általában nem áll fenn, így szükség van az áteresztőképesség mérésére terhelt esetben is. Ez a terhelés az egyszerre több folyam áthaladása az adott switch-en. Összegezve ennek a tesztnek a célja az OpenFlow switch áteresztőképességének mérése terhelés mellett (pl. UDP adatfolyam).

Mérési elrendezés egy 5 portos OpenFlow switch esetében:



4. ábra: Áteresztőképesség terhelt környezetben

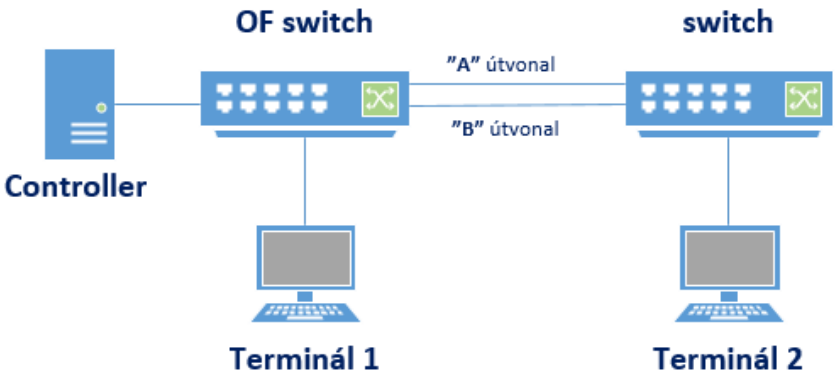
A fenti ábrán látható elrendezésben a zöld adatfolyam átviteli sebességét mérjük úgy, hogy közben a piros útvonalon terhelés folyik, ami lehet például egy UDP adatfolyam.

Természetesen ugyanúgy, mint az előző pontban ez a teszt is felbontható további altesztekre a switch állapotától függően.

Átkapcsolási teszt

Elméletben és az OpenFlow-t bemutató prezentációkban azt ígérik, hogy a protokoll gyors beavatkozásra képes dinamikus, gyakran változó hálózati környezetben.

Új folyamok tanulásának és kapcsolásának igénye akár egy futó adatkapcsolat közben is felmerülhet. De milyen hatásai vannak ennek a váltásnak? Van-e csomagvesztés, hatással van-e a késleltetésre és a csomagok megérkezésének sorrendjére? Ezeknek a kérdéseknek a megválaszolásához véleményünk szerint elengedhetetlen az *Átkapcsolási teszt* elvégzése az ellenőrizni kívánt eszközökön. Bonyolultságuktól függően számos ilyen teszt létezik. Mi az 5. ábrán látható egyszerű tesztrendszert építettük ki, mely jól átlátható eredményt ad egy flow-mod üzenet érvényesítésének átvitelre történő hatásáról:



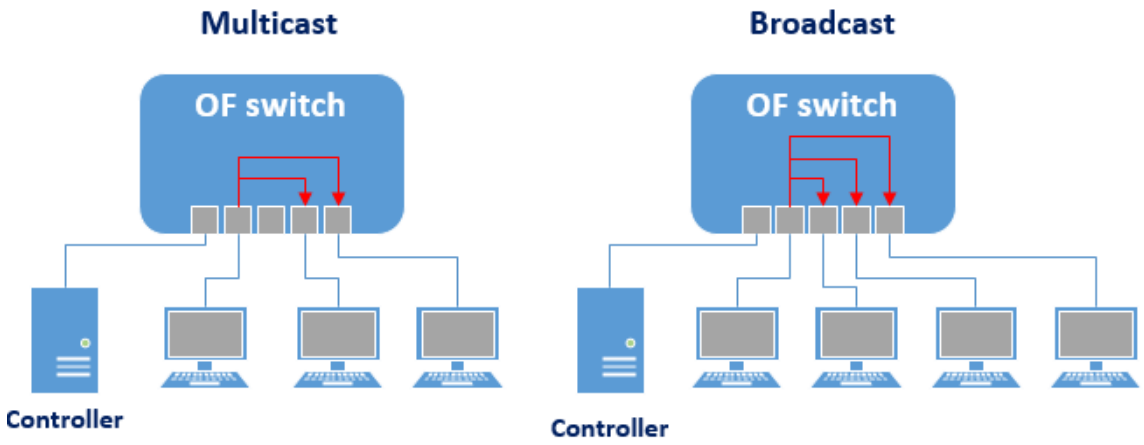
5. ábra: Átkapcsolási topológia

A példában két terminál gép közötti *A* útvonalon történő átvitel tulajdonságait vizsgáljuk, miközben flow-mod parancsokkal a folyam útvonalát a *B* útvonalra módosítjuk.

Összegezve ennek a tesztnek a végcélja egy folyam sebességének, késleltetésének, csomagvesztésének és csomagok sorrend helyességének a vizsgálata, ilyen módon átfogó képet adva az eszköz dinamikusságáról.

Multicast / Broadcast teljesítmény

Mivel számos hálózati protokoll és alkalmazás lételeme a broadcast vagy multicast az adott hálózatban, azaz a beérkező csomagok továbbítása az összes vagy a kiválasztott portokon keresztül, így elengedhetetlen a switch tesztelése ebből a szempontból is. Mivel a broadcast vagy multicast esetében a bejövő csomagot többszörözzük és egyszerre több porton való továbbítását végezzük, így egyáltalán nem triviális, hogy a csomag továbbítási sebessége azonos a már előző pontokban mért áteresztőképességgel. Egy-egy broadcast és multicast csomagtovábbítási példa egy 4 portos switch-en a 6. ábrán látható:



6. ábra: Multicast és Broadcast topológia

Itt is, mint ahogy a többi mérésnél számos egyéb beállítás és switch állapot befolyásolhatja az eszköz teljesítményét (pl. különböző táblaméretek, különböző bejegyzések, aktív/passzív portok száma, stb.), emiatt a teljes körű eredmény eléréséhez több teszt elvégzésére van szükség.

3.3. MÉRÉSCSOPORTOK EREDMÉNYEINEK ÉRTELMEZÉSE

Természetesen még számtalan egyéb tesztet lehetne elvégezni minden kategóriában, azonban álláspontunk szerint ezek voltak a legjelentősebbek mivel ezeknek elvégzése bárki számára széleskörű ismeretet ad az adott termék OpenFlow támogatottságáról és ebből következően SDN hálózatban történő alkalmazhatóságáról is.

Végezetül vegyük végig az SDN hálózat tervezésének és kivitelezésének egyes lépéseit, ill. hogy ezekben milyen segítséget nyújt az általunk kialakított és ajánlott módszertan:

Egy vállalat részéről először megszületik az ötlet, hogy SDN hálózatot kíván felépíteni. Ekkor - ahogy azt már az előzőekben kifejtettük - feltétlenül szükség van előzetes ismeretekre a jövőbeli hálózatban működő eszközökről, mivel a gyártófüggetlenség mára még nem valósult meg (teljesen). Ezek az információk elengedhetetlenek a hálózattervezők számára, hiszen ennek ismeretében kell majd programozniuk a vezérlési síkban működő egy vagy több kontrollert.

Véleményünk szerint az előzetes információk közül a következőek elengedhetetlenek:

1. switch által támogatott táblák száma
2. folyambejegyzés mentéséhez szükséges lépések
3. teli tábla jelzés támogatása
4. beérkező csomag hardveres feldolgozás-kritériumainak ismerete

Második feladat, a kontroller és a hálózat számára a lehető legjobb teljesítményű eszközök kiválasztása. A vezérlő részére értékes információkat az *Elemi OpenFlow műveletek teljesítmény tesztelése* eredményei nyújtanak. Például a hálózat méretétől függően kulcsfontosságú az elmenthető bejegyzések száma, az egy másodperc alatt feldolgozott flow-mod üzenetek száma pedig erősen befolyásolja a hálózat dinamikusságát.

E tekintetben a következő Openflow specifikus adatokat tartjuk kulcsfontosságúaknak:

5. különböző táblákba menthető folyambejegyzések száma
6. flow-mod/másodperc értéke
7. packet-in csomagok másodpercenkénti száma

Miután leszűkült a kör az OpenFlow specifikus kritériumoknak megfelelő switch-ekre, harmadik lépésként a hálózat szempontjából egy adott árkategóriában a lehető legjobb teljesítményű eszköz kiválasztása a feladat.

Az általunk fontosnak gondolt vizsgálandó teljesítmények a következőek:

8. áteresztőképesség teljesítménye ideális és terhelt esetben
9. UDP adatfolyam mellett, csomagvesztés és sorrendhelyesség a folyambejegyzések módosítása esetén
10. áteresztőképesség teljesítménye Multicast és Broadcast működésnél

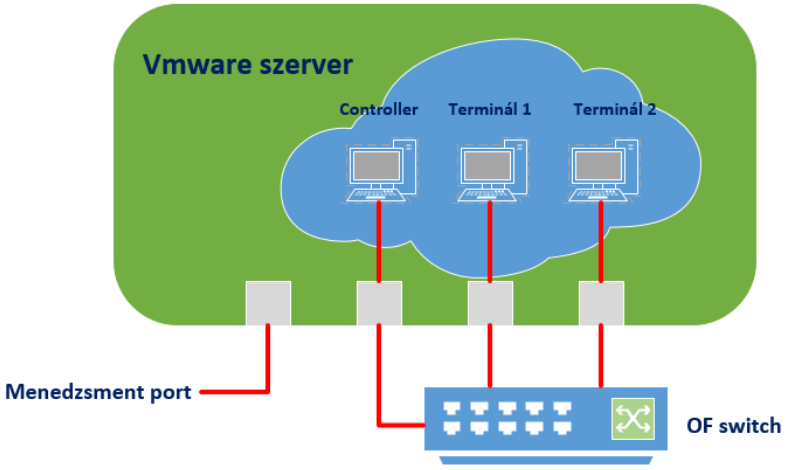
Ha mindezek tudatában vagyunk, akkor valós, tiszta képet kapunk a vásárolni kívánt termékekről, ezek alapján elkezdődhet a hálózatépítés.

A bemutatott módszertan alkalmazásának demonstrálására a dolgozat következő fejezetében az általunk elérhető switch-ek tényleges tesztelését és összehasonlítását mutatjuk be.

4. VALÓS SWITCH-EK TESZTJEI

4.1. LABORKÖRNYEZET

Munkánk során a BME I épület IE318-as számú HSN laborban dolgoztunk. A laborban nem rendelkeztünk "hivatalos mérőeszközökkel", így célunk volt egy olyan tesztkörnyezet kialakítása, mely a lehető legpontosabb eredményeket képes produkálni. Rendelkezésünkre állt egy IBM x3550 M4 szerver, melyen a VMware ESXi 5.5.0 rendszer futott. Ez egy bare-metal hypervisor, aminek segítségével virtuális gépeket (Virtual Machine, VM) tudunk létrehozni. A szerver nagy kapacitású erőforrásai miatt célravezetőnek láttuk VM-ek használatát, a laborban található PC-k helyett. Létrehoztunk három virtuális gépet (kontroller, terminál1 és terminál2), amikre 64 bites Ubuntu 14.04 operációs rendszert és csak a számunkra fontos szoftvereket telepítettük, hogy minimalizáljuk a tesztekre ható háttérben futó folyamatok számát. Az IBM szerver négy Gigabit Ethernet porttal rendelkezik, amelyek közül a menedzsment porton keresztül érjük el magát a szervert. A másik hármat kiosztottuk az általunk létrehozott VM-eknek, hogy ezeken keresztül kapcsolódjunk az OpenFlow switch-ekhez. Az így kialakított mérési elrendezés 7. ábrán látható.



7. ábra: Laborkörnyezet

Ma már számos OpenFlow kontroller létezik mind-mind különböző teljesítménnyel. Mivel mi az OpenFlow switch-ek teljesítményét voltunk hivatottak tesztelni és nem pedig a kontrollerét, így ezt az elemét a hálózatnak nem használtuk a tesztjeinkben (ellenkező esetben az eredmények összefüggésben lettek volna az éppen használt kontroller típusával). Azonban egy OpenFlow switch folyambejegyzések nélkül nem továbbítja a csomagokat, így minden mérés előtt a szükséges bejegyzéseket manuálisan vagy shell scriptekkel töltöttük fel a Controller nevű VM-en elérhető parancssori `ovs-ofctl` program segítségével, ami az Open vSwitch[9] (OVS) programcsomag része [13]. A VM-re telepített OVS verziószáma: 2.3.0. Méréseink során nagy hasznát vettük a WireShark forgalomanalizáló program 1.12-es

[9] Az OVS egy nyílt forráskódú virtuális multilayer switch, amelynek legújabb verziója (2.3.0) már támogatja az OpenFlow 1.3-as verzióját is.

verziójának, amely képes feldolgozni az 1.0 és az 1.3 verziónak megfelelő OpenFlow üzeneteket is [14].

A feladatunk a laborban lévő, több gyártótól származó, switch-ek SDN hálózatban való megfelelésének tesztelése volt. A rendelkezésünkre álló eszközök a következőek:

- Cisco Catalyst 3750-X 24TL
- Juniper EX2200-24T-4G
- Arista 7048T-A
- MikroTik RB750GL
- MikroTik RB2011iLS-IN
- HP 3500-24G-PoE+ yl

4.2. ÁLTALÁNOS MEGÁLLAPÍTÁSOK

Amennyiben egy switch-et SDN hálózatban akarunk használni, akkor az első feladat az OpenFlow támogatottságának vizsgálata. Habár azt gondolnánk, hogy az eszköz használati útmutatóját kinyitva minden OpenFlow specifikus kérdésre választ kaphatunk, a gyakorlatban ez egyáltalán nem ennyire triviális.

Cisco és Juniper

A laborban lévő Cisco és Juniper switch-ek közül egyik sem támogatja hardveres feldolgozással az OpenFlow-t. A Junipernél felvettük a kapcsolatot a vállalattal, hogy létezik-e OpenFlow kompatibilis firmware az eszközre. Válaszban kaptuk, hogy még fejlesztés alatt áll, így ennél az eszköznél kizártuk az OpenFlow kompatibilitás tesztelését. A Cisco esetében is hasonlóan jártunk, ennél sem támogatja a jelenlegi firmware az OpenFlow protokollt, csak a Cisco saját SDN megvalósítását.

Arista 7048T-A

A laborban lévő Arista switch az EOS 4.12.7.1 firmware-rel rendelkezik. Az Arista eszközök szériái különböző hardver chip-ekkel (ASIC) rendelkeznek, melyek közül a Trident támogatja az OpenFlow protokollt. A laborban lévő switch ettől különböző, Petra chippel rendelkezik. Habár hivatalos megoldás nincs, azért lehetőségünk van erre az eszközre is ezt a funkcionalitást implementálni. Mivel az Arista Extensible Operating System (EOS) Linux alapú hálózati operációs rendszer, így felvetődött a kérdés, hogy lehetséges-e az Open vSwitch futtatása az eszközön. Következtetésképpen a feladat egy OVS telepítése volt az Arista 7048T-A eszközre.

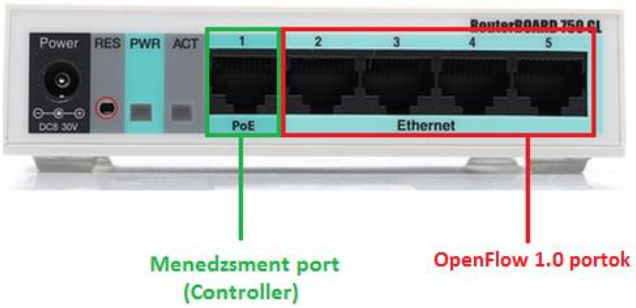
Mivel az Arista switch operációs rendszere sok szempontból korlátozva van a helyes működés biztonsága érdekében, így számos nehézség adódott az OVS telepítése során:

- tiltva van a telepítés az eszközön
- felhasználó által végzett módosítások a file directory-ban minden reboot esetén törlődnek
- portok nem csatlakoztathatóak közvetlenül az Open vSwitch-hez

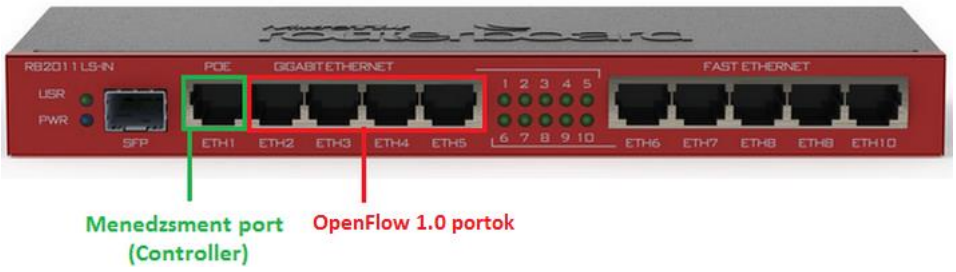
A fentebbi problémák megoldásával végül sikerült létrehozni egy user space módban futó, OpenFlow-t támogató Open vSwitch-et. Első lépésként teszteltük az áteresztőképességet két terminál gép segítségével. A végeredmény ~3 Mbits/sec lett 2,8%-os CPU felhasználás mellett. Mivel ez a teljesítmény rendkívül gyenge egy ilyen kaliberű eszközhöz képest, így a továbbiakban nem foglalkoztunk az Arista switch OpenFlow kompatibilitásának ellenőrzésével.

MikroTik RB 750GL és RB 2011

Habár a MikroTik eszközök legújabb firmware verziója (v3.18, RouterOS 6.18) már tartalmazza az OpenFlow 1.0-ás protokollt, a hivatalos weblapjukon felhívják a figyelmet, hogy jelen pillanatban ez a funkció még nem "production ready", azaz adódhatnak problémák a használata során. Ennek ellenére, véleményünk szerint a MikroTik-ek érdekes összehasonlítási alapot szolgáltathatnak az ezektől nagyságrendekkel drágább switch-ekkel szemben. Ahogy az az eszköz alacsony árából is következtethető, ezek a switch-ek csak szoftveres feldolgozást képesek biztosítani az OpenFlow folyamai számára. Nem tartalmaznak TCAM memóriát és mindkettő Atheros 8327 chippel rendelkezik. Az RB 2011-es switch-ben kettő darab is található, mivel az egyik a gigabites portokat, a másik pedig a 100 megabites portokat kezeli. RAM memóriájuk azonos méretű (64 MB). Fontos különbség a kettő között a processzor teljesítményében található, ugyanis a kisebb 400 MHz-es míg a nagyobb 600MHz-es CPU-val rendelkezik. A firmware frissítés után az alábbi módon konfiguráltuk az eszközöket:



8. ábra: MikroTik 750GL portkiosztás



9. ábra: MikroTik RB2011 portkiosztás

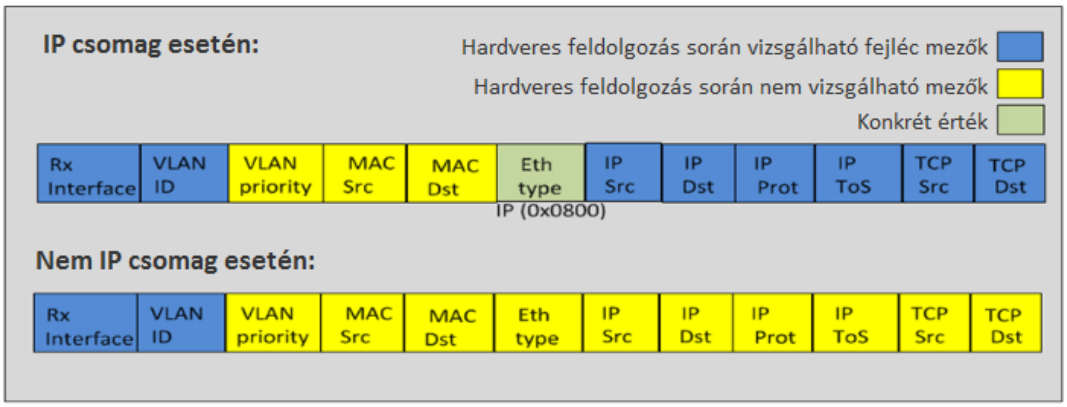
Mivel a mai igényeknek megfelelő (gigabites kapcsolatok) működést kívántuk tesztelni, így az RB 2011-es MikroTik esetében a lassabb portokkal nem foglalkoztunk, nem konfiguráltuk őket a működéshez. Végeredményként kaptunk 2 db 4 porttal rendelkező OpenFlow 1.0-ás switch-et.

HP 3500 yl

A K.15.15.0006 firmware-rel rendelkező HP 3500 switch már támogatja mind az 1.0.0-ás mind az 1.3.1-es OpenFlow verziót. Fontos kiemelnünk, hogy ennél az eszköznél mind hardveres, mind szoftveres feldolgozás is megvalósul.

Ha egy folyam szoftveresen programozott, akkor nem érhető el vonali sebesség a csomagtovábbítás során. Ahogy azt már korábban említettük, a HP 3500 esetében is egy hagyományos switch-csel van dolgunk. Mint ilyen, szűk keresztmetszetet jelent az OpenFlow feldolgozáshoz megfelelő programozható memóriák mérete. Ebből adódóan néhány esetben a HP mérnökei nem tudták megoldani a folyamok gyors, hardveres kapcsolását a jelenlegi hardver architektúra fizikai korlátai miatt.

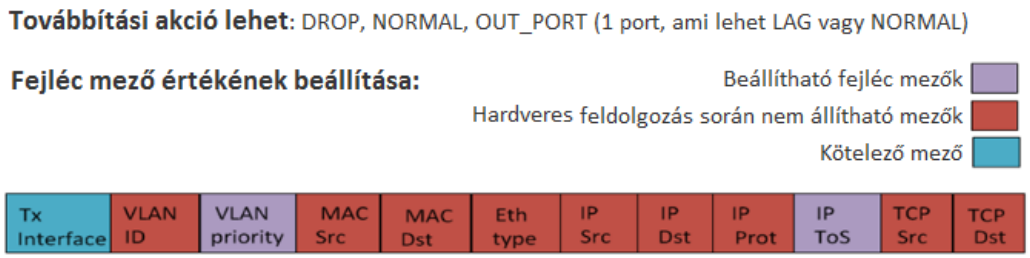
Fejlécmező egyezés vizsgálat során a következő szabályok vonatkoznak a feldolgozásra:



10. ábra: Hardveres feldolgozás kritériumai fejlécvizsgálatkor

Az illeszkedés vizsgálat után az adott bejegyzésben meghatározott akciók hajtódnak végre. Az akció általában adott porton történő továbbítást jelent, de lehet egyes fejléc mezők

megváltoztatása is. A HP switch esetén hardveresen elvégezhető akciókra a következő megkötések vonatkoznak:



11. ábra: Hardveres feldolgozás kritériumai akciók esetén

Tehát a következő illeszkedési feltételek esetén a folyambejegyzések, egy speciális táblába, az úgynevezett szoftveres folyamtáblába fognak kerülni:

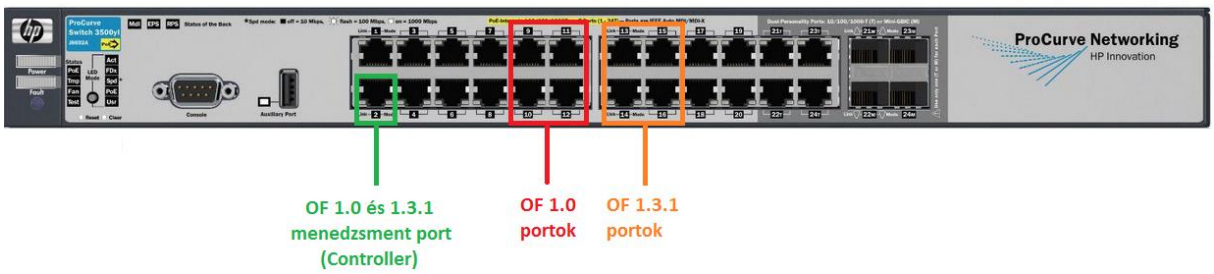
- a forrás vagy cél MAC cím specifikált
- a VLAN-PCP mező specifikált
- az EtherType mező „nem-IP” típusú

Továbbá hardveres feldolgozás használatakor engedélyezett akciók a következőek:

- kimenő interfész megadása (kötelező)
- VLAN prioritás mező beállítása
- IP TOS mező beállítása

A szoftveres feldolgozás lassabb kapcsolást feltételez, továbbá a nagy mennyiségű illeszkedő csomag az eszköz processzorának túlterheléséhez is vezethet. Emiatt a gyártó egy felső korlátot is meghatározott az ilyen módú feldolgozásra. Ezt a szoftveres feldolgozási rátát OpenFlow instance-onként[10] lehet beállítani, csomag/másodperc léptékkel 0 és 10.000 között.

Mivel a MikroTik switch-ek 4 portosak, így a hiteles összehasonlítás érdekében a HP-ban létrehozott OpenFlow instance-okat is 4 kezelhető porttal láttuk el:



12. ábra: HP portkiosztás

[10] Egymástól független, OpenFlow paraméterekkel meghatározható működési tartományok.

Összegezve a 6 switch közül, csupán csak három kompatibilis az OpenFlow protokollal olyan mértékben, hogy érdemes legyen további elemző és összehasonlító vizsgálatokat végezni rajtuk.

A tesztkörnyezet kialakítása után kezdtük el az előző fejezetben ismertetett módszertan lépéseit elvégezni a tesztelni kívánt eszközökön.

4.3. HP – OpenFlow 1.0 tesztelése

Elsőnek a HP switch-re implementált OpenFlow 1.0-ás instance-t (innentől HP 1.0) teszteltük. Vegyük végig ennek a módszertan lépései szerint kapott eredményeit!

4.3.1. Alapvető OpenFlow működési tesztek

Első lépésként megvizsgáltuk a switch (a kontroller helyes beállításához elengedhetetlen) előzetes tudnivalóit.

Mivel az instance az OpenFlow 1.0-ával dolgozik, így a támogatott táblák száma nem is volt kérdés, ugyanis az a protokollba foglaltak szerint maximum egy lehet.

Ezután a működés alapjaként szolgáló folyambejegyzés mentését volt célszerű ellenőrizni, hiszen ez elengedhetetlen a csomagtovábbítás helyes működéshez. Tesztünkben egy wildcard bejegyzést küldtünk az `ovs-ofctl` program segítségével a switch számára, miközben a Controller VM-re telepített WireShark-kal monitoroztuk a megfelelő interfészeket. Az eredmény várható volt, azaz a `flow-mod` és a `Barrier Request` üzenetek elküldése után sikeres válaszként kaptuk meg a `Barrier Reply`-t. Ezután manuálisan is ellenőriztük a mentés helyességét az `ovs-ofctl dump-flows` parancs (bejegyzések listázása) segítségével. A kimenetben látható volt az általunk küldött folyambejegyzés, tehát a mentés sikeresen megtörtént.

Ezt követően teszteltük a teli tábla jelzés meglétét. Az `ovs-ofctl add-flows` parancs fájlból olvassa be a switch-nek küldeni kívánt folyambejegyzéseket, ezért elsődleges feladatunk egy olyan bash script megírása volt, amely ezeket a bejegyzéseket tartalmazó fájlt létrehozza. Miután ezzel végeztünk kezdődhetett a feltöltés. Eredményként tapasztaltuk, hogy a táblák feltöltődése után egy `OFPT_ERROR : OFPFMFC_TABLE_FULL` üzenetet küld a switch a kontroller számára, vagyis a HP mérnökei implementálták a termékükbe ezt az ajánlott funkciót. Végül utánajártunk a switch hardveres és szoftveres továbbításhoz szükséges kritériumainak. Ez a laborkörnyezet pont általános megállapításaiban már részletezésre került.

Összefoglalva a kapott eredményeket:

Támogatott táblák száma:	1
Bejegyzések mentése:	OK
Teli tábla jelzés:	Implementálva
Hardveres feldolgozás kritériumai:	A kritériumok a 10. és 11. ábrán láthatóak

4.3.2. ELEMI OPENFLOW MŰVELETEK TELJESÍTMÉNY TESZTJEI

Amint elvégeztük a legfontosabb alapvető teszteket, a következő lépés az OpenFlow belső működésének vizsgálata volt.

Táblakapacitás teszt

Arra a kérdésre már választ kaptunk, hogy mennyi folyamtáblát támogat a HP 1.0, azonban arra még nem, hogy hány folyambejegyzés fér egy táblában. A TCAM-mel rendelkező switch-ek esetében fontos különbséget tenni a különböző típusú bejegyzések táblába mentései között. Az OpenFlow 1.0-ás tábla a RAM és TCAM memóriaterületeket logikailag együtt kezeli. Amíg van szabad hely a TCAM memóriában, addig oda tölti fel a bejegyzéseket, hiszen innen tudja garantálni a hardveres feldolgozást. Ha már nincs több szabad tárhely, csak akkor kezdi a bejegyzések RAM-ba való mentését. Ahhoz, hogy külön mérni tudjuk a TCAM méretét, olyan bejegyzéseket kell feltölteni, melyek megfelelnek a hardveres feldolgozás kritériumainak. Ellenkező esetben a switch kizárólag a RAM memóriában tárol.

Ennek tudatában a Controller VM-ről addig töltöttük fel a folyambejegyzéseket, amíg a HP 1.0 teli tábla jelzéssel nem válaszolt. Ekkor egy `dump-tables` paranccsal ellenőriztük a tábla bejegyzéseinek számát. A kapott eredmény 65536 db bejegyzés volt. Ez azonban még nem adott választ minden kérdésünkre, hiszen tudjuk, hogy ez egy logikai tábla, ami összefogja a különböző memóriaterületeket. Ahhoz, hogy kiderítsük a TCAM és RAM memóriákba mentett bejegyzések számát a HP switch-re belépve lekérdeztük az instance tulajdonságait. A kapott eredmény meglepő lett:

```
HP3# show openflow instance test10
No. of Hw Flows           : 760        // TCAM
No. of Sw Flows           : 64776      // RAM
```

A HP specifikációiban 1500-at említenek, mint a modulonként hardveresen feldolgozható bejegyzések száma. A különbségre magyarázatot ad, hogy korlátozott az OpenFlow-nak ajánlható TCAM memória mennyisége, mivel ezt az erőforrást az eszköz más funkciói is használják. Azonban a felhasznált mennyiség állítható. Ez alapértelmezetten 50%

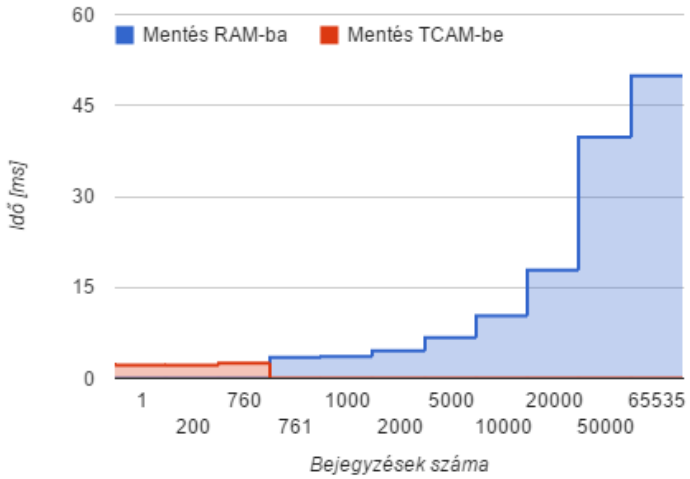
– ebből adódik a kapott 760-as eredmény –, de 0 és 100% között dinamikusan, a használattól függően változtatható.

Fontos megemlítenünk, hogy a HP switch-en konfigurált OpenFlow instance-ok egymással versenyeznek a bejegyzések tárolására közösen használt memóriaterületekért. Emiatt, ha az egyik instance abból már felhasznált bizonyos mennyiséget, akkor a másik már annyival kevesebb bejegyzést képes elmenteni.

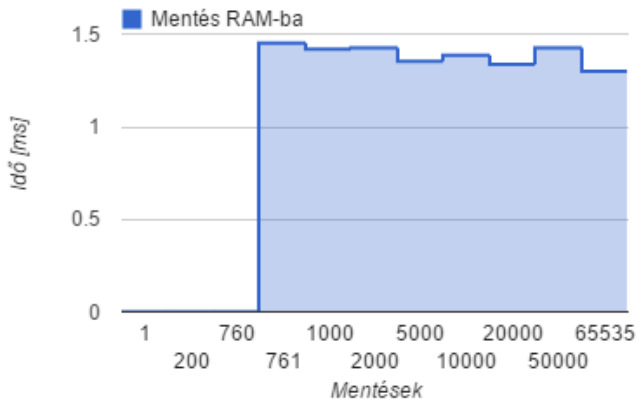
Flow-mod teljesítmény teszt

Azt már tudjuk, hogy mekkora a HP 1.0 táblájának kapacitása. Joggal merül fel a kérdés, hogy mennyi idő a bejegyezések elmentése, illetve törlése. Erre ad választ a flow-mod teljesítmény teszt eredménye. Ez a mérés több, különböző fajta altesztet foglal magába. Ezek az altesztek az elmentett bejegyzések számának hatását vizsgálják a mentéshez, módosításhoz vagy törléshez szükséges időre, wildcard vagy exact bejegyzések esetén.

Tesztjeink eredményeit a következő diagramok írják le:



13. ábra: HP 1.0 - Wildcard mentési idő



14. ábra: HP 1.0 - Exact mentési idő

Ahogy a 13. ábrán látható a wildcard bejegyzések esetében, ahogy egyre több elem került a táblába, úgy növekedett a szükséges mentési idő is. Exact bejegyzéseknél ez nem így történt. Erre a magyarázat a memóriakezelésben keresendő.

A teli táblák (65536 db) bejegyzéseinek törlési ideje exact bejegyzések esetén **2,09 másodperc** volt. Wildcard bejegyzések esetén a törlési időt nem tudtuk kiszámolni, mivel a kiadott parancs után a switch összeomlott és újraindult.

A wildcard bejegyzések TCAM memóriában történő mentésekor átlagosan körülbelül **420 flow-mod/másodperc** sebesség érhető el. Amint a RAM memória feltöltése kezdődik el ez az érték hirtelen lecsökken **280 flow-mod/másodperc** értékre. A tábla feltöltődése közben ez folyamatosan csökken, míg 20.000 bejegyzés esetén például csak **55 flow-mod/másodperc**, addig a szinte teljesen telített tábla estén már csak **20 flow-mod/másodperc.** Ezzel szemben az exact match bejegyzések mentésének ideje **750-800 flow-mod/másodperc** körül alakul a tábla feltöltöttségétől függetlenül.

Packet-in teszt

Ebben a tesztben kellően nagy mennyiségű csomagot generálunk és küldünk az OpenFlow switch egy portjára. Ezek olyan csomagok, amelyek nem illeszkednek a folyamtábla egy bejegyzésére sem. Erre válaszul a switch packet-in üzeneteket generál és küld a kontroller számára, amiket Wireshark-kal rögzítettünk, majd szkriptek segítségével feldolgozunk. Végül megkapjuk, hogy másodpercenként hány packet-in csomagot képes generálni az eszköz. Nagy mennyiségű csomag generálásához az nping [15] nevű programot alkalmaztuk. Használata során megfelelő paraméterezése mellett 10.000 csomagot generáltunk másodpercenként.

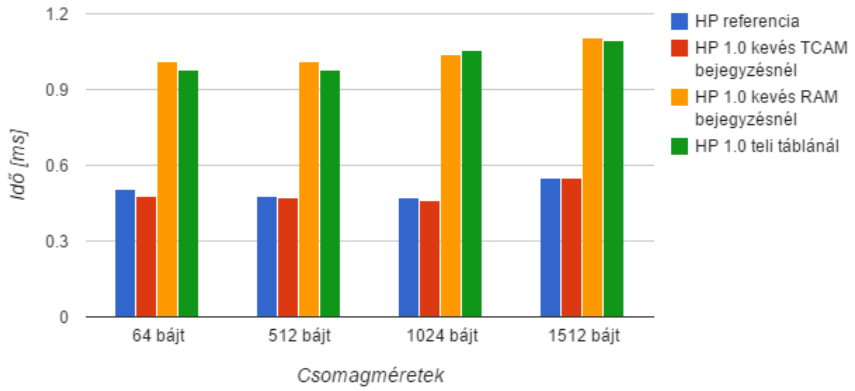
Mivel a feldolgozás sebességére hatással van a tábla feltöltöttsége, ezért üres, 30 ezer és 60 ezer – egyébként irreleváns – bejegyzéssel feltöltött tábla esetén is végeztünk méréseket.

A HP 1.0-nál az elért **packet-in/másodperc** metrika üres tábla esetén átlagosan **5100**, 30 ezer bejegyzéssel feltöltött esetben **5000**, míg 60 ezer bejegyzésnél **4300** körül alakul.

4.3.3. MAGASSZINTŰ HÁLÓZATI TELJESÍTMÉNY TESZTEK

RTT mérés

Az RTT alapján mért OpenFlow feldolgozás overhead-jét az 15. ábrán láthatjuk.

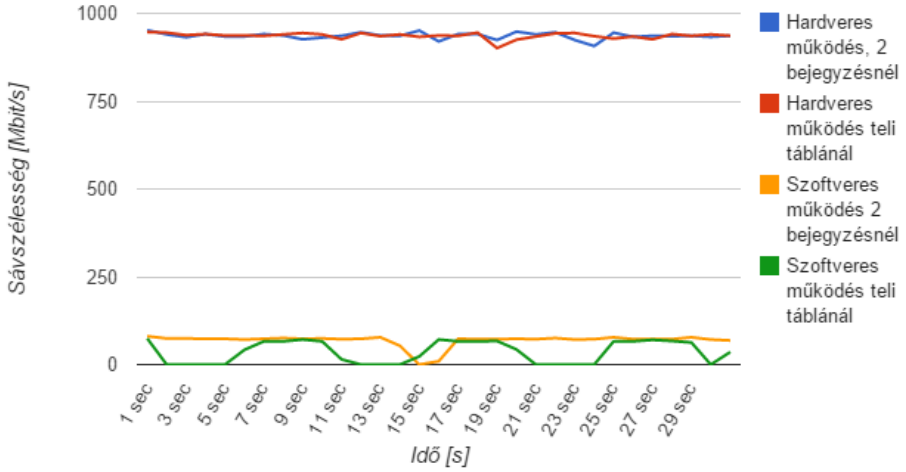


15. ábra: HP 1.0 - RTT idő

Látható, hogy a szoftveres feldolgozáskor, vagyis mikor egy bejegyzés RAM memóriában tárolódik lényegesen (több mint 0,5ms-mal) lassabb a körülfordulási idő.

Áteresztőképesség mérése egy folyam esetén

Áteresztőképesség méréssel teszteltük az OpenFlow feldolgozás sebességét két terminál közötti TCP és UDP forgalom esetén. Hogy hiteles eredményt kapjunk több különböző switch beállítás mellett végeztük el a méréseket. Külön vizsgáltuk hardveres / szoftveres feldolgozás és üres / teli tábla esetén. Az eredmények az alábbi diagramon láthatóak:



16. ábra: HP 1.0 - TCP áteresztőképesség

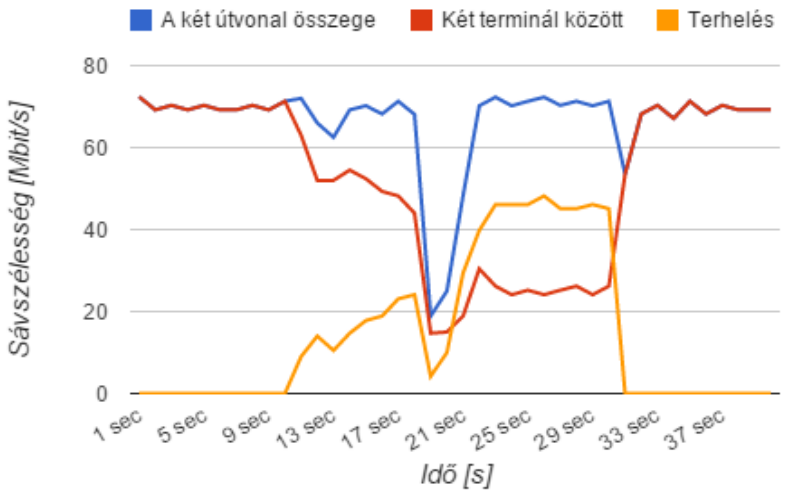
A 16. ábrán látható, hogy TCP folyamnál hardveres feldolgozás esetén **~940 Mbit/s** érhető el. Ennél nagyságrenddel lassabb a szoftveres feldolgozás sebessége, ami jobb esetben is csak **~75 Mbit/s.**

UDP protokoll esetén az átlageredmények a következőek lettek:

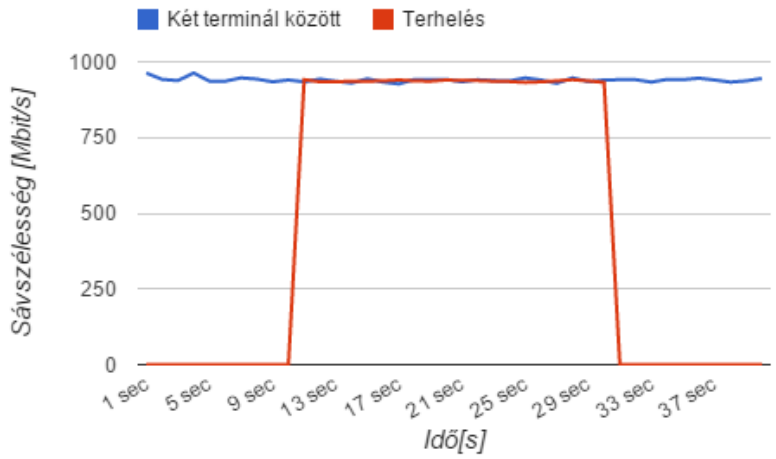
Hardveres feldolgozás két bejegyzés esetén:	**802 Mbit/s**
Hardveres feldolgozás teli tábla esetén:	**797 Mbit/s**
Szoftveres feldolgozás két bejegyzés esetén:	**78,7 Mbit/s**
Szoftveres feldolgozás teli tábla esetén:	**30,2 Mbit/s**

Áteresztőképesség mérése terhelt környezetben

Az előző sebességteszteket mind ideális esetben hajtottuk végre, vagyis a switch-nek csak két portja volt aktív a kapcsoláskor. Érdemes azonban extra terhelt állapotban is tesztelni az átvitelt. Erre hoztuk létre a 4. ábrán látható rendszert. Itt is csak úgy, mint terhelés nélkül, többféle beállítás mellett mértük az áteresztőképességet a két terminál között. Az eredmények a következőek lettek:



17. ábra: HP 1.0 - Szoftveres áteresztőképesség terhelt esetben



18. ábra: HP 1.0 - Hardveres áteresztőképesség terhelt esetben

Az átlagok szoftveres feldolgozás esetén **48,6 Mb/s** a két terminál között és **27,8Mb/s** a terhelő gépek között. Hardveres feldolgozásnál **939 Mb/s** a két terminál között és **937Mb/s** a terhelő gépek között.

UDP protokoll esetén a szoftveres feldolgozásnál **63,9 Mbit/s** és **45,2 Mbit/s** sávszélesség volt mérhető a két terminál és a terhelő gépek között. Hardveres feldolgozásnál nem volt számottevő különbség a sávszélességben, mind a terminálok mind a terhelések között **798 Mbit/s** volt.

Az eredményekből látható, hogy a hardveres feldolgozás esetén nincs hatással az áteresztőképességre a switch egyéb portjain történő forgalom. Ez viszont már nem mondható el a szoftveres feldolgozásra, ahol az áteresztőképesség kb. 70 Mbit/s a folyamok számától függetlenül, tehát ezen a sávszélességen osztozik az összes szoftveres feldolgozású folyam.

Átkapcsolási teszt

Felépítettük az 5. ábrán látható hálózatot. A méréshez UDP kapcsolatot (pl.: videostream, VoIP) létesítettünk a két terminálgép között az *'A'* útvonalon. Minden ötödik másodpercben ezt az útvonalat megváltoztattuk (flow-mod modify üzenet) a vele párhuzamosra. Ahhoz, hogy a kapott értékek hibamentesek legyenek, az UDP kapcsolat 100 Mbit/s-el működött, mivel ekkor alap esetben 0% volt a csomagvesztés, és mindegyik csomag helyes sorrendben érkezett meg. Ezekhez a referenciaértékekhez hasonlítottuk a kapott eredményt. Az eredmény a 2. táblázatban látható.

2. táblázat: HP 1.0 - Átkapcsolási veszteségek

	csomagveszteség	**sorrendi hiba**
1. folyammódosítás	14/8548 (0,16%)	50
2. folyammódosítás	13/8547 (0,15%)	50
3. folyammódosítás	12/8546 (0,14%)	48
4. folyammódosítás	8/8547 (0,094%)	42

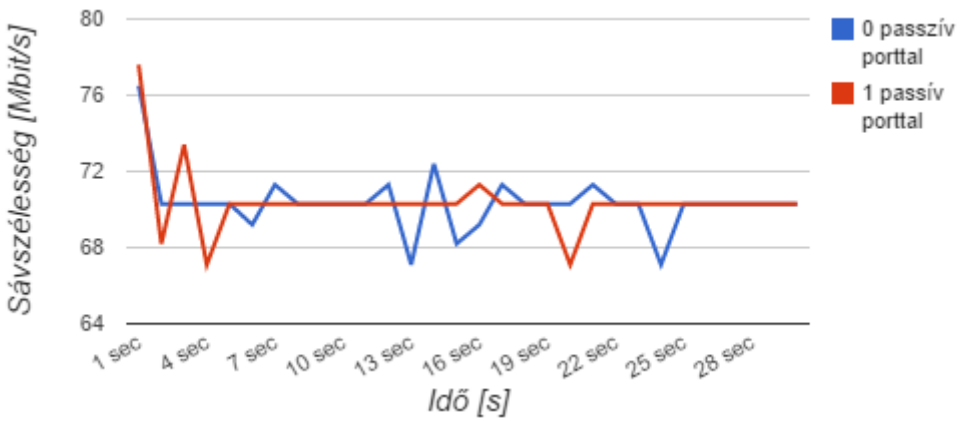
Összességében látható, hogy a 100 Mbit/s-os UDP kapcsolat esetében a módosító flow-mod üzenet bár csak kis mértékben, de befolyásolta a folyamot.

Multicast / Broadcast teljesítmény

Itt a kérdés, hogy egyszerre több porton történő csomagtovábbítás esetén az eszköz vajon megközelíti-e az áteresztőképesség tesztnél mért sebességét. A HP 1.0 esetében a válasz egyértelműen nem. Ennek oka, hogy ha egy folyambejegyzés akciójában több portra történő továbbítás van beállítva, akkor az erre illeszkedő beérkező csomagokat a HP kizárólag szoftveresen tudja feldolgozni. Fontos megemlíteni, hogy az aktív portok számától függően több tesztet végeztünk el ebben a kategóriában. A kapott eredmények alább láthatóak:

19. ábra: HP 1.0 - Broadcast teszt

A 19. ábrán látható, hogy az átlag sávszélességek az **54,4 Mb/s** (0 passzív port), **54,8 Mb/s** (1 passzív port) és **54,4 Mb/s** (2 passzív port) voltak.



20. ábra: HP 1.0 - Multicast teszt

Az átlag sávszélességek az **70,4 Mb/s** (0 passzív port) és **70,4 Mb/s** (1 passzív port) voltak.

<u>UDP protokoll esetén az átlageredmények a következőek lettek:</u>

Broadcast továbbítás passzív port nélkül:	**53,2 Mbit/s**
Broadcast továbbítás egy passzív porttal:	**55,5 Mbit/s**
Broadcast továbbítás két passzív porttal:	**55,5 Mbit/s**
Multicast továbbítás passzív port nélkül:	**71,1 Mbit/s**
Multicast továbbítás egy passzív porttal:	**71,2 Mbit/s**

Ezzel a teszttel elvégeztük a módszertanunkban definiált mindhárom tesztkategória vizsgálatát, így átfogó képet kaptunk a HP 1.0 switch OpenFlow kompatibilitásáról és működéséről. Természetesen még számtalan egyéb tesztet lehetne elvégezni minden kategóriában, azonban álláspontunk szerint ezek voltak a legjelentősebbek.

Maradt még három vizsgálandó eszközünk (HP OpenFlow 1.3 instance, MikroTik RB750GL, MikroTik RB2011iLS-IN), ezért elvégeztük azok tesztelését is. Mivel a mérési rendszerek és lépések ugyanazok voltak, mint a HP 1.0 esetében, így a tesztek eredményeit felsorolásszerűen írtuk csak le, viszont az érdekes és figyelemre méltó tapasztalatokat továbbra is bővebben kifejtjük.

4.4. HP – OpenFlow 1.3 tesztelése

A rövidítés érdekében a HP OpenFlow instance 1.3-at innentől HP 1.3-nak nevezzük.

4.4.1. Alapvető OpenFlow működési tesztek

Támogatott táblák száma:	5+1
Bejegyzések mentése:	OK
Teli tábla jelzés:	Implementálva
Hardveres feldolgozás kritériumai:	A kritériumok a 10. és 11. ábrán láthatóak

A HP 1.3 összesen 5+1 folyamtáblát támogat. Ebből egy a hardveres bejegyzéseknek (TCAM) és négy a szoftveres bejegyzéseknek (RAM) lett implementálva. A szoftveres táblák száma 0 és 4 között használattól függően állítható. Létezik egy 0 ID-jű tábla is (+1), mely a pipeline feldolgozás működését biztosítja. Ebben csak alapértelmezett bejegyzés lehet, manuálisan nem tudjuk a tartalmát módosítani. Az összes többi szempontból a switch megegyezik a HP 1.0-val ebben a kategóriában.

4.4.2. Elemi OpenFlow műveletek teljesítmény tesztjei

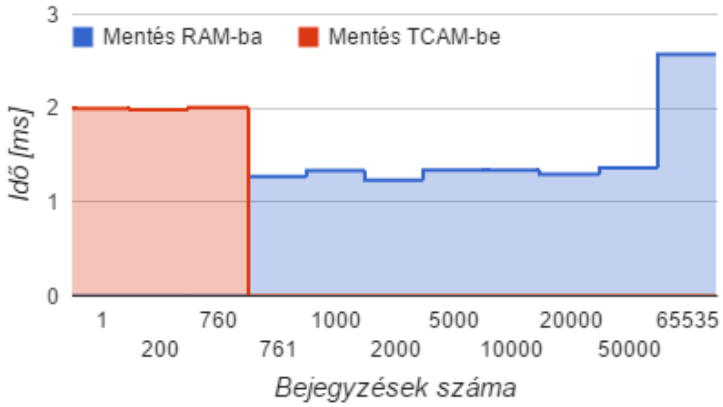
Táblakapacitás teszt

A HP 1.3 nagy sebességű táblája (ID100) a TCAM memóriába leképezett. Ide csak a hardveres feldolgozás kritériumainak megfelelő wildcard bejegyzések kerülhetnek. Ennek a kapacitása maximum 1516 bejegyzés. Ezen kívül létezik még a "szoftver" tábla (ID200), mely a RAM memóriába menti a bejegyzéseket. Az ide került bejegyzésekre illeszkedő csomagokat a HP 1.3 csak szoftveresen tudja feldolgozni. Ennek a táblának a mérete 65565 bejegyzés - TCAM-ben lévő bejegyzések. További opció, hogy lehetőség van ezt a szoftvertáblát feldarabolni, több (maximum 4) kisebb táblára. Amennyiben ezt megtesszük, akkor nincs szükség külön a méretük definiálására, mivel a táblák egymással dinamikusan versenyeznek az elérhető szabad memóriáért.

Flow-mod teljesítmény teszt

Mivel ez a switch az OpenFlow 1.3-at támogatja, külön tartalmaz hardvertáblát és szoftvertáblát, emiatt ezeknek a feltöltési sebességét külön kellett mérnünk. Esetünkben a TCAM memória 50%-át állítottuk be az OpenFlow számára, így a maximum bejegyzés, amit tárolni tudott 762 darab volt. A feltöltése során nem volt meghatározó különbség a bejegyzések mentésének idejében, vagyis az első és az utolsó bejegyzés mentése is hasonló időbe tellett (kb. 2,0 ms, **500 flow-mod/másodperc**). A 762 bejegyzés törlése **2,41 másodpercet** vett igénybe.

Meglepő módon itt a RAM memóriába helyezett bejegyzések mentési ideje alacsonyabb volt, mint a TCAM esetében. Nagyjából a tábla telítettségétől függetlenül 1,3 ms körül alakult (**750-800 flow-mod/másodperc**), csak a maximális tábla feltöltöttség közelében növekedett 2,6 ms körülire. A 65ezer bejegyzés törlése **23,49 másodpercet** vett igénybe.



21. ábra: HP 1.3 - Wildcard mentési idő

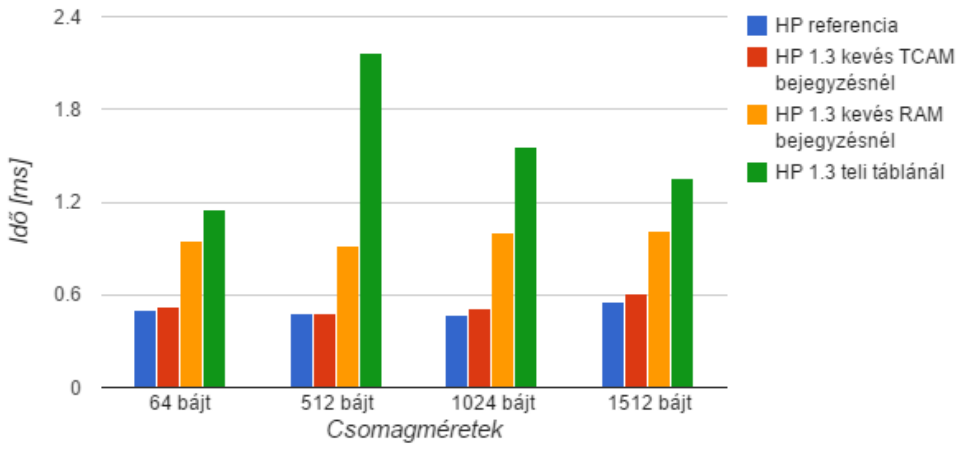
A HP 1.0-val ellentétben itt nem volt különbség wildcard és exact bejegyzések feltöltésének ideje között.

Packet-in teljesítmény teszt

A HP 1.3-nál elért **packet-in/másodperc** metrika üres tábla esetén átlagosan **1600**, 30 ezer bejegyzéssel feltöltött esetben **800**, míg 60 ezernél **100** körül alakul.

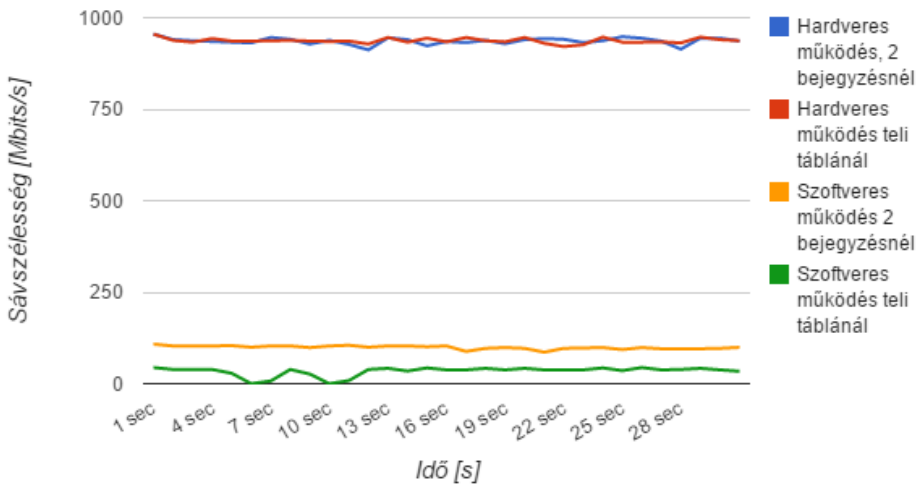
4.4.3. MAGASSZINTŰ HÁLÓZATI TELJESÍTMÉNY TESZTEK

RTT mérés



22. ábra: HP 1.3 - RTT idő

Áteresztőképesség mérése 1 folyam esetén



23. ábra: HP 1.3 - TCP áteresztőképesség

<u>Átlagok:</u>

Hardveres működésnél: **936,3 Mb/s** és **937,1 Mb/s**

Szoftveres működésnél: **99,2 Mb/s** és **33,9 Mb/s**

<u>UDP protokoll esetén az átlageredmények a következőek lettek:</u>

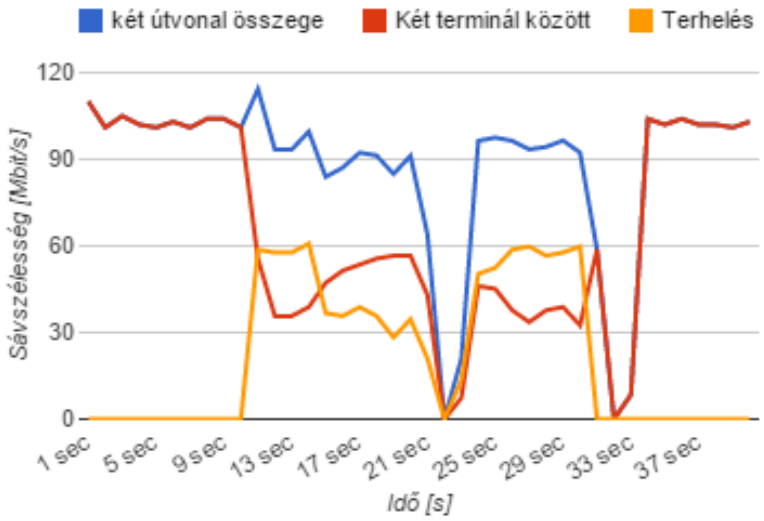
Hardveres feldolgozás két bejegyzés esetén: **800 Mbit/s**

Hardveres feldolgozás teli tábla esetén: **799 Mbit/s**

Szoftveres feldolgozás két bejegyzés esetén: **99 Mbit/s**

Szoftveres feldolgozás teli tábla esetén: **38,8 Mbit/s**

Áteresztőképesség mérése terhelt környezetben



24. ábra: HP 1.3 - Szoftveres áteresztőképesség terhelt esetben

25. ábra: HP 1.3 - Hardveres áteresztőképesség terhelt esetben

A 24. ábrán, a szoftveres feldolgozás átlagaik **65,5 Mb/s** a két terminál között és **43,7 Mb/s** a terhelő gépek között. Hardveres feldolgozás esetén **942 Mb/s** a két terminál között és **935 Mb/s** a terhelő gépek között.

UDP protokoll esetén az átlageredmények a következőek lettek:

Hardveres feldolgozás esetén a két terminál között:	**803 Mbit/s**
Hardveres feldolgozás esetén a "terhelő" gépek között:	**800 Mbit/s**
Szoftveres feldolgozás esetén a két terminál között:	**70,9 Mbit/s**
Szoftveres feldolgozás esetén "terhelő" gépek között:	**76,7 Mbit/s**

Átkapcsolási teszt

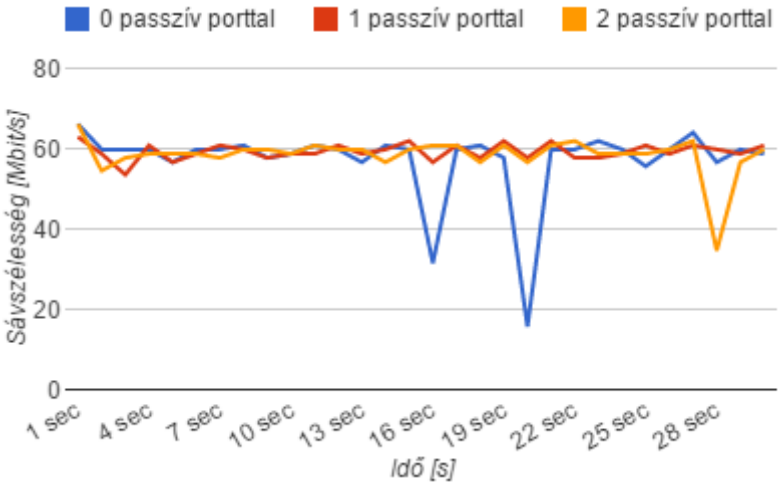
Az útvonal módosító bejegyzést minden ötödik másodpercben léptettük érvénybe. Az így kapott eredmények egy 100 Mbit/s sávszélességű UDP adatkapcsolat esetén:

3. táblázat: HP 1.3 - Átkapcsolási veszteségek

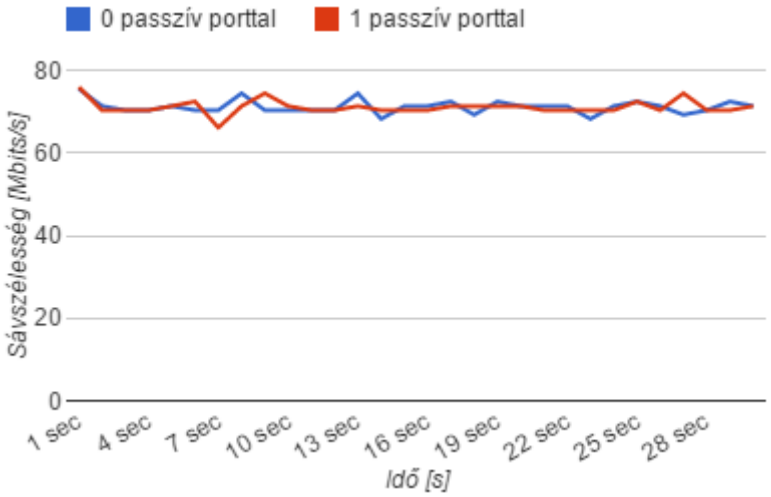
	csomagveszteség	sorrendi hibák
1. folyammódosítás	11/8547 (0,13%)	40
2. folyammódosítás	11/8546 (0,13%)	39
3. folyammódosítás	11/8547 (0,13%)	39
4. folyammódosítás	12/8552 (0,14%)	39

Minden további időpontban az összes csomag helyes sorrendben érkezett meg és csomagveszteség is 0% volt.

Multicast / Broadcast teljesítmény



26. ábra: HP 1.3 - Broadcast teszt



27. ábra: HP 1.3 - Multicast teszt

A broadcast átlag sávszélességek az **51,3 Mb/s** (0 passzív port), **59,2 Mb/s** (1 passzív port) és **58,5 Mb/s** (2 passzív port) voltak. A multicast átlag sávszélességek pedig az **71,1 Mb/s** (0 passzív port) és **71,1 Mb/s.**

UDP protokoll esetén az átlageredmények a következőek lettek:

Broadcast továbbítás passzív port nélkül:	**72 Mbit/s**
Broadcast továbbítás egy passzív porttal:	**72 Mbit/s**
Broadcast továbbítás két passzív porttal:	**71,5 Mbit/s**
Multicast továbbítás passzív port nélkül:	**94,6 Mbit/s**
Multicast továbbítás egy passzív porttal:	**94,5 Mbit/s**

4.5. MikroTik 750GL – OpenFlow 1.0 tesztelése

4.5.1. Alapvető OpenFlow működési tesztek

Támogatott táblák száma:	1
Bejegyzések mentése:	Csak kontroller jelenlétében
Teli tábla jelzés:	Nincs implementálva
Hardveres feldolgozás kritériumai:	Nincs hardveres feldolgozás

Ebben a teszt kategóriában akad néhány fontos megjegyezni való a MikroTik 750GL esetében, ugyanis működése eltér az eddig megismert HP eszközökétől. Mikor egy bejegyzés mentését vizsgáltuk, a Wireshark-kal mért üzenetváltások alapján minden a működésnek megfelelően történt.

Azonban amikor ezek után kilistáztuk a switch folyambejegyzéseit, meglepő eredmény fogadott minket. A `Barrier-reply` üzenettel nyugtázott bejegyzés mentése nem történt meg, az eszköz folyamtáblája üres volt. Ahelyett, hogy a flow-mod bejegyzés után a MikroTik 750GL hibaüzenettel válaszolt volna, hogy nem mentette el a bejegyzést, még meg is erősítette a végbemenetelét. Abban az esetben, ha egy kontrollert csatlakoztattunk a switch-hez (esetünkben POX-ot), ezek után már gond nélkül működött a manuálisan történő (nem a POX kontroller által) bejegyzés feltöltés. Fontos tudnivaló, hogy a POX kontrollert működő alkalmazás nélkül indítottuk el, vagyis kizártuk azt, hogy az a tudomásunk nélkül küldjön folyambejegyzéseket a switch számára. Ezek a hibák elkerülése végett a további tesztek esetében is így történt.

A teli tábla jelzés tesztnél még egy hibára figyeltünk fel. A mérés elvégzése után eredményül kaptuk, hogy a MikroTik nem jelzi a kontroller számára, hogy megtelt a folyamtáblája. A folyamatos bejegyzés feltöltés egy idő után megszakadt, ami egy TCP reset üzenet miatt következett be.

Ez a switch nem tartalmaz TCAM memóriát, így egyértelmű volt, hogy a bejegyzéseket az eszköz RAM (64MB) memóriába menti. A feltöltés után ezt megvizsgálva láttuk, hogy körülbelül 35 MB szabad tárhelye van még. Feltételezésünk az volt, hogy a switch előre kiosztott valamennyi memóriaterületet az OpenFlow számára, ami a feltöltés során megtelt. Ennek ellenére mikor újraindítottuk a bejegyzések feltöltését (az előzőek törlése nélkül), az eszköz egy nincs szabad memória hibaüzenet küldése helyett folytatta a bejegyzés mentést. A folyambejegyzések feltöltése egy bizonyos idő után mindig megszakadtak, azonban egy újabb elindítás esetén folytatódtak. Ezután addig ismételtük a feltöltéseket, amíg a RAM memória elfogyott. Ekkor a switch hibaüzenet küldése helyett összeomlott és újraindult.

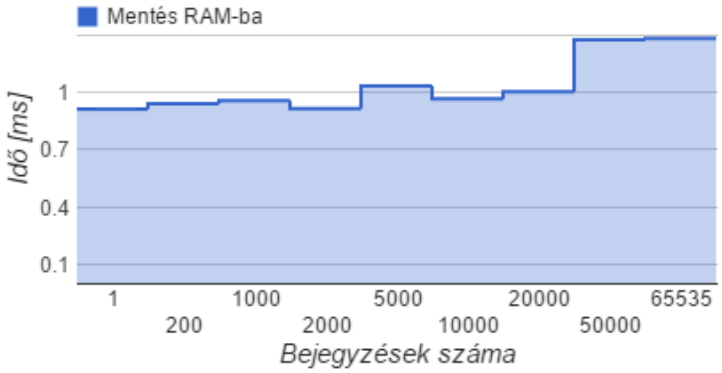
4.5.2. ELEMI OPENFLOW MŰVELETEK TELJESÍTMÉNY TESZTJEI

Táblakapacitás teszt

Ahogy azt az előző pontban részleteztük, a bejegyzés feltöltés periodikusan egy bizonyos idő után megszakadt. Egy-egy ilyen ciklus során a feltöltött folyambejegyzések száma mindig különböző volt. A worst-case esetet nézve a switch 29.585 db (6,2 MB) wildcard bejegyzést és 26.390 db (5,6 MB) exact bejegyzést tudott feltölteni. A maximális szám, amit az összeomlás előtt el tudtunk érni, körülbelül 108.000 bejegyzés.

Flow-mod teljesítmény teszt

Kezdetben, a kevés bejegyzéssel feltöltött tábla esetén körülbelül **1000 flow-mod/másodperces** mentési sebesség érhető el. Később, mikor a folyamtábla egyre jobban feltöltődik ez az érték **800 flow-mod/másodperc** sebesség alá csökken.



28. ábra: MikroTik 750GL - Wildcard mentési idő

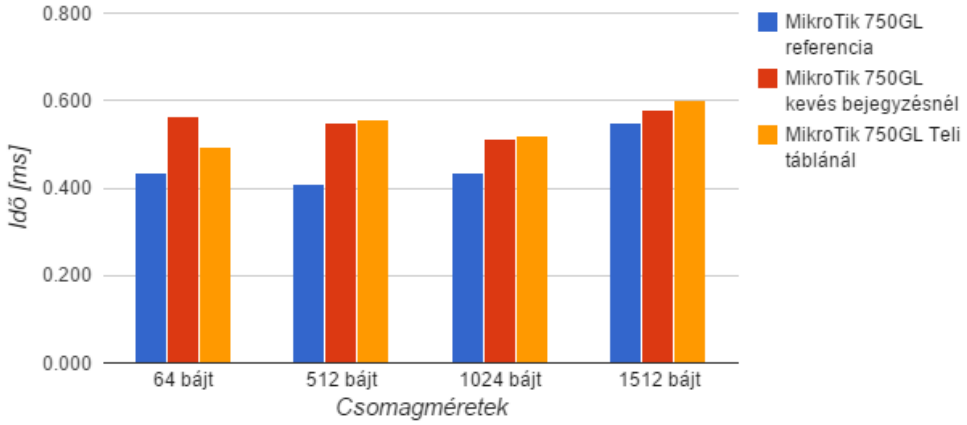
Packet-in teszt

Meg kell említeni, hogy a MikroTik egy kontrollernek küldött packet-in csomagban egyszerre több ismeretlen feldolgozású, beérkezett csomagot is becsomagol.

Az összességében elért **packet-in/másodperc** metrika üres tábla esetén átlagosan **9600**, 30 ezer bejegyzéssel feltöltött esetben **180**, míg 60 ezernél **150** körül alakul.

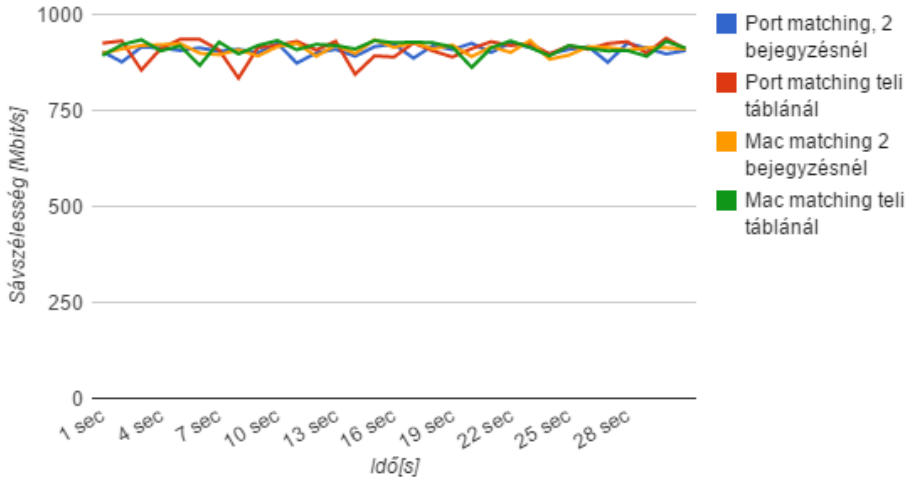
4.5.3. MAGASSZINTŰ HÁLÓZATI TELJESÍTMÉNY TESZTEK

RTT mérés



29. ábra: MikroTik 750GL - RTT idő

Áteresztőképesség mérése 1 folyam esetén



30. ábra: MikroTik 750GL - TCP áteresztőképesség

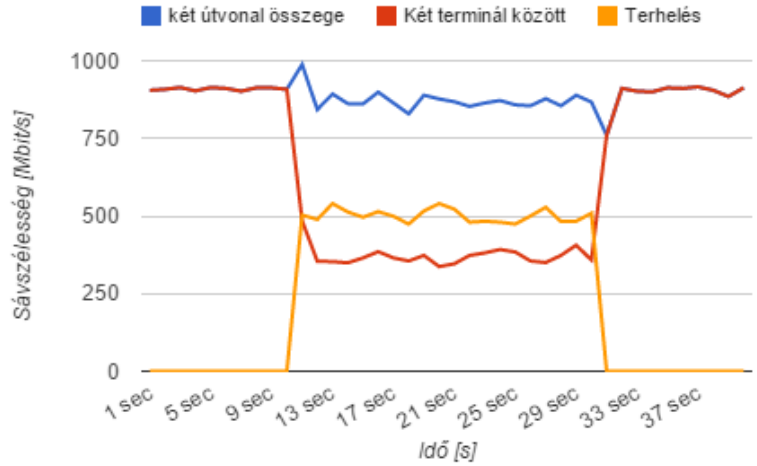
Átlagok:

Port szűrésnél:	**906 Mb/s** és **909 Mb/s**
MAC cím szűrésnél:	**910 Mb/s** és **912 Mb/s**
POX által feltöltött exact match esetén:	**909 Mb/s** és **911 Mb/s**

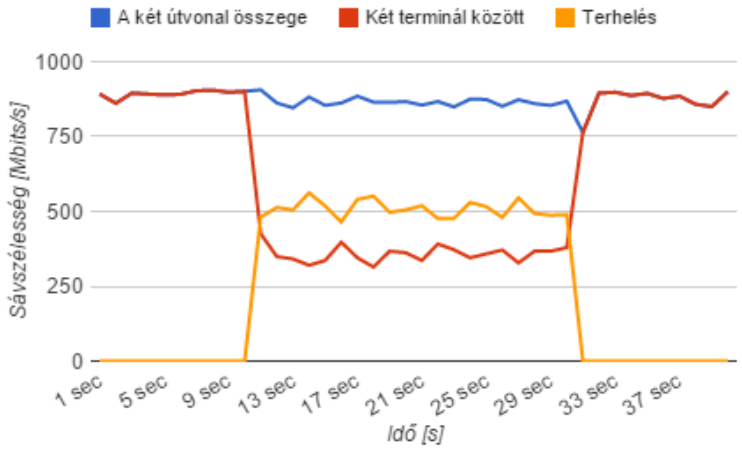
UDP protokoll esetén az átlageredmények a következőek lettek:

Bejövő port vizsgálat, két bejegyzés esetén:	**796 Mbit/s**
Bejövő port vizsgálat, teli tábla esetén:	**797 Mbit/s**
MAC cím vizsgálat, két bejegyzés esetén:	**799 Mbit/s**
MAC cím vizsgálat, teli tábla esetén:	**796 Mbit/s**

Áteresztőképesség mérése terhelt környezetben



31. ábra: MikroTik 750GL - Terhelt áteresztőképesség port vizsgálat esetén



32. ábra: MikroTik 750GL - Terhelt áteresztőképesség MAC cím vizsgálat esetén

Az átlagok port vizsgálat esetén **620 Mb/s** a két terminál között és **507 Mb/s** a terhelő gépek között. MAC cím vizsgálatnál **636 Mb/s** a két terminál között és **501,2Mb/s** a terhelő gépek között.

<u>UDP protokoll esetén az átlageredmények a következőek lettek:</u>

Bejövő portvizsgálat esetén a két terminál között:	**625 Mbit/s**
Bejövő portvizsgálat esetén a “terhelő” gépek között:	**462 Mbit/s**
MAC cím vizsgálat esetén a két terminál között:	**625 Mbit/s**
MAC cím vizsgálat esetén “terhelő” gépek között:	**470 Mbit/s**

Átkapcsolási teszt

Az útvonal-módosító bejegyzést minden ötödik másodpercben léptettük érvénybe. Az így kapott eredmények egy 100 Mbit/s sávszélességű UDP adatkapcsolat esetén:

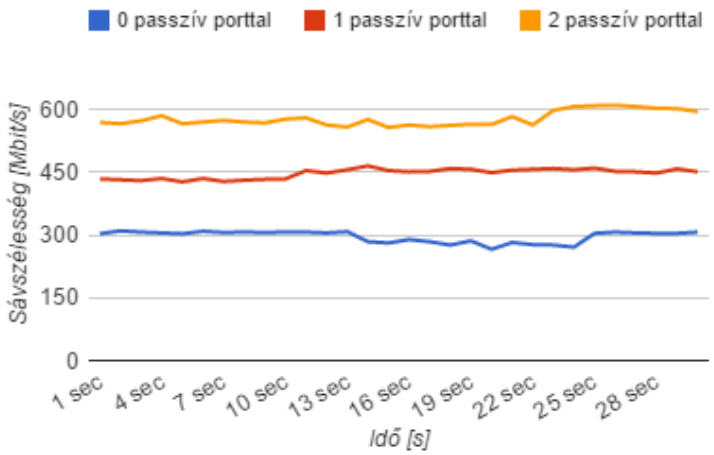
4. táblázat: MikroTik 750GL - Átkapcsolási veszteségek

	csomagveszteség	**sorrendi hibák**
1. folyammódosítás	3/8548 (0,035%)	0
2. folyammódosítás	3/8546 (0,35%)	0
3. folyammódosítás	3/8547 (0,35%)	0
4. folyammódosítás	4/8548 (0,47%)	1

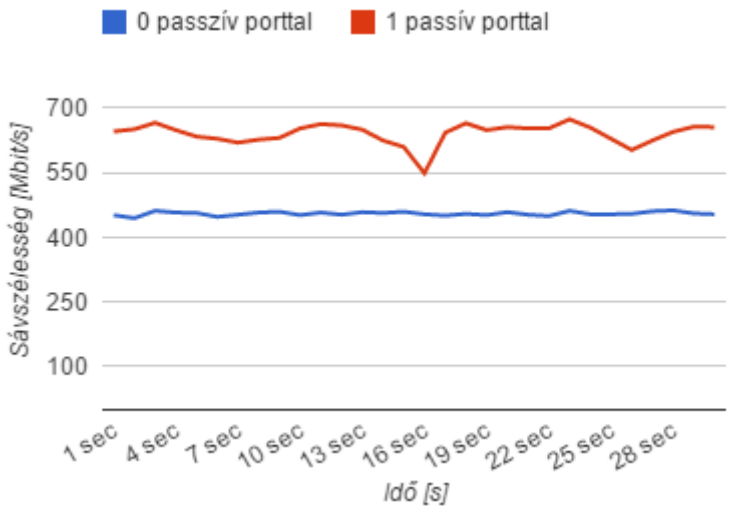
Minden további időpontban az összes csomag helyes sorrendben érkezett meg és csomagveszteség is 0% volt.

Érdekességképpen kipróbáltuk az átvitelt 200 Mbit/s sávszélességű UDP kapcsolattal is. Ekkor egy újabb hibát tapasztaltunk az eszközön. Ahelyett, hogy nagyobb csomagveszteséggel működött volna, egyszerűen csak összeomlott a switch OpenFlow instance-a. Tehát minden funkciója működött kivéve, ami az OpenFlow-hoz volt kapcsolható. Ekkor kizárólag az újraindítás segített a további működés eléréséhez.

Multicast / Broadcast teljesítmény



33. ábra: MikroTik 750GL - Broadcast teszt



34. ábra: MikroTik 750GL - Multicast teszt

A 33. ábrán látható átlag sávszélességek az **296 Mb/s** (0 passzív port), **447 Mb/s** (1 passzív port) és **578 Mb/s** (2 passzív port) voltak a broadcast esetén. Multicastnál **455 Mb/s** (0 passzív port) és **637 Mb/s** sávszélességek voltak.

UDP protokoll esetén az átlageredmények a következőek lettek:

Broadcast továbbítás passzív port nélkül:	**310 Mbit/s**
Broadcast továbbítás egy passzív porttal:	**432 Mbit/s**
Broadcast továbbítás két passzív porttal:	**571 Mbit/s**
Multicast továbbítás passzív port nélkül:	**459 Mbit/s**
Multicast továbbítás egy passzív porttal:	**639 Mbit/s**

4.6. MikroTik RB2011 – OpenFlow 1.0 tesztelése

4.6.1. Alapvető OpenFlow működési tesztek

Ebben a tesztcsoport kategóriában a switch minden szempontból a megegyezik az előző pontban tárgyalt, MikroTik RB750 GL tulajdonságaival.

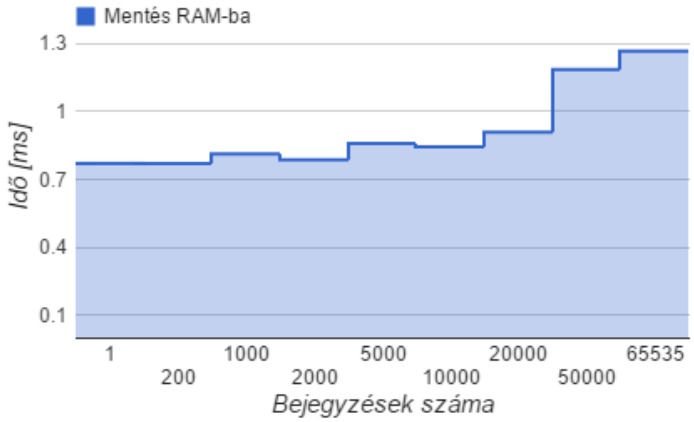
4.6.2. Elemi OpenFlow műveletek teljesítmény tesztjei

Táblakapacitás teszt

Ez a MikroTik eszköz sem tartalmaz TCAM memóriát, így minden bejegyést a RAM-jában tárol. A hibajelenség, miszerint a folyamatos folyambejegyzés feltöltés periódikusan közel azonos időnként megszakad itt is jelen volt. Egy megszakadás előtt a feltöltésre kerülő folyambejegyzések száma mindig különböző volt. A worst case esetet nézve a switch 34452 db wildcard bejegyzést (7,5 MB) és 30.387 db (6,5MB) exact bejegyzést

tudott feltölteni. Ugyanakkor többszöri feltöltés esetén nagyobb érték is elérhető, míg meg nem telik a RAM.

Flow-mod teljesítmény teszt



35. ábra: MikroTik RB2011 - Wildcard bejegyzések mentési ideje

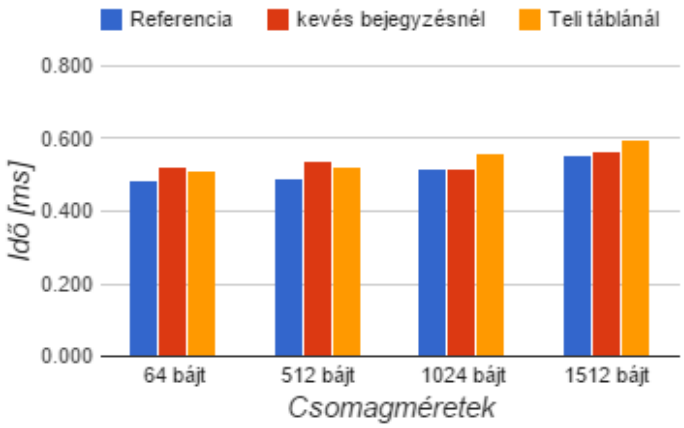
Itt sem volt lényeges különbség az exact és a wildcard bejegyzések mentése között. Kezdetben a mentési sebesség nagyjából **1200-1300 flow-mod/másodperc** körül alakult. Nagymértékben feltöltött tábla esetén **800 flow-mod/másodperc** körüli értékre csökken.

Packet-in teszt

Ez a swich is a kontrollernek küldött packet-in csomagba egyszerre több ismeretlen feldolgozású csomag fejléceit is becsomagolja. Az összességében elért **packet-in/másodperc** metrika üres tábla esetén átlagosan **9600**, 30 ezer bejegyzéssel feltöltött esetben **150**, míg 60 ezernél is **150** körül alakul.

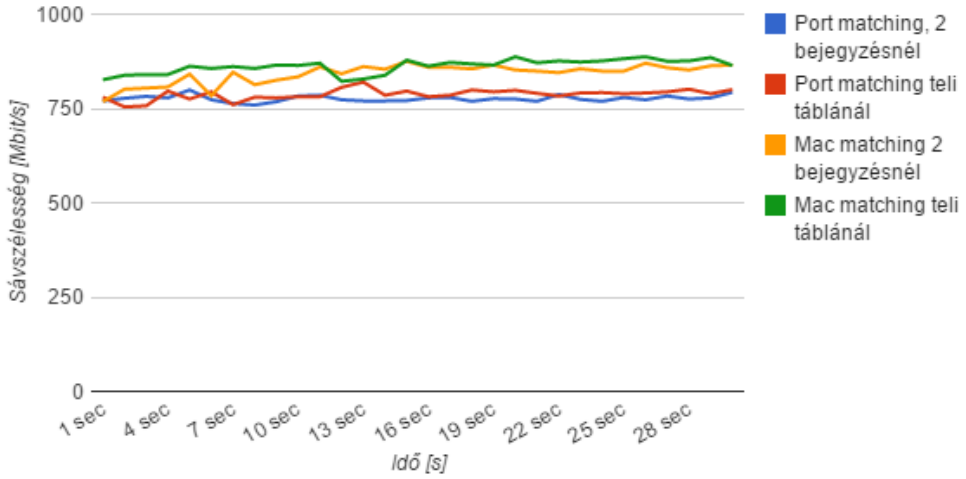
4.6.3. MAGASSZINTŰ HÁLÓZATI TELJESÍTMÉNY TESZTEK

RTT mérés



36. ábra: MikroTik RB2011 - RTT idő

Áteresztőképesség mérése 1 folyam esetén



37. ábra: MikroTik RB2011 - TCP áteresztőképesség

TCP Átlagok:

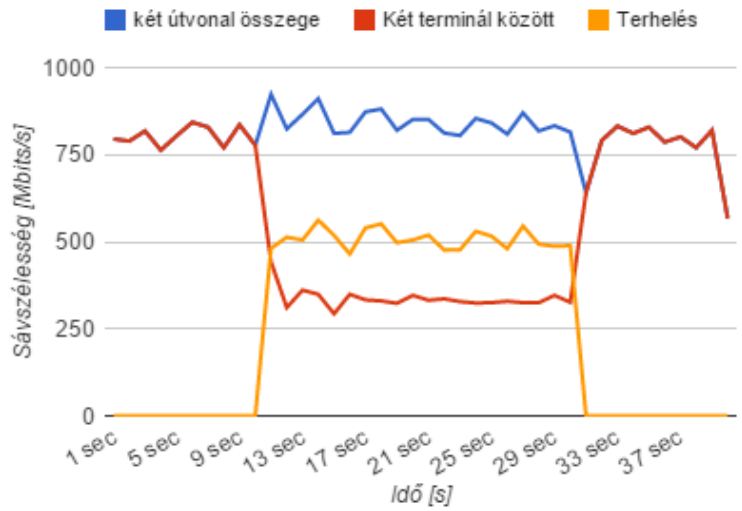
Port szűrésnél:	**776 Mb/s** és **787 Mb/s**
MAC cím szűrésnél:	**842 Mb/s** és **862 Mb/s**
POX által feltöltött exact match esetén:	**866 Mb/s**

Fontos tudnivaló, hogy a mért eredmények közben a CPU kihasználtság kb. 48-80%-on volt. Látható, hogy mivel szoftveres feldolgozásról van szó, a CPU működés határozza meg a switch éppen aktuális áteresztőképességét. A fenti ábrán az átlagteljesítményt rögzítettük. Érdekes eredmény, hogy a switch teli tábla esetén jobban teljesített. Az így kapott eredménykor a tábla wildcard bejegyzésekkel volt feltöltve, tehát adódik a kérdés, hogy vajon ugyanígy viselkedik az eszköz exact bejegyzések esetén is? A válasz nem. Az így kapott sávszélesség 866 Mbit/s volt 75%-os CPU kihasználtság mellett.

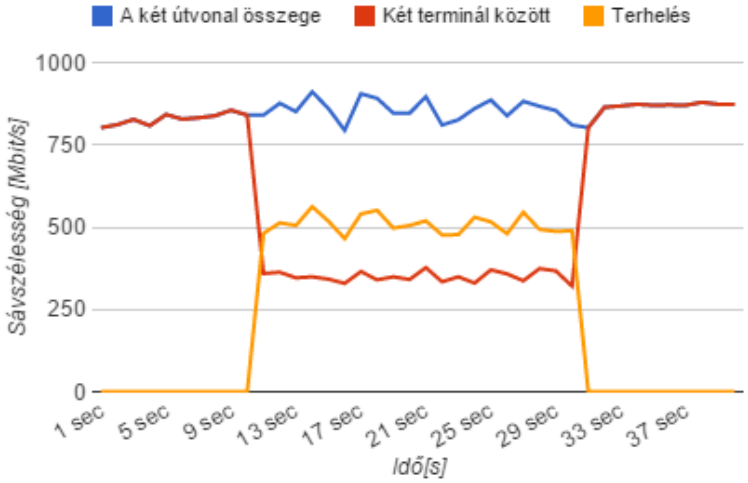
UDP protokoll esetén az átlagos eredmények a következőek lettek:

Port szűrésnél:	**753 Mbit/s** és **752 Mbit/s**
MAC cím szűrésnél:	**771 Mbit/s** és **799 Mbit/s**
Exact bejegyzés esetén:	**806 Mbit/s**

Backplane kapacitás tesztelés



38. ábra: MikroTik RB2011 - Terhelt áteresztőképesség port vizsgálat esetén



39. ábra: MikroTik RB2011 - Terhelt áteresztőképesség MAC cím vizsgálat esetén

A port vizsgálatnál a két terminál között sávszélesség átlaga **636 Mb/s** és terhelő gépek között **507 Mb/s**. MAC cím vizsgálatnál **598 Mb/s** a két terminál között és **507 Mb/s** a terhelő gépek között.

<u>UDP protokoll esetén az átlageredmények a következőek lettek:</u>

Bejövő portvizsgálat esetén a két terminál között:	**612 Mbit/s**
Bejövő portvizsgálat esetén a "terhelő" gépek között:	**455 Mbit/s**
MAC cím vizsgálat esetén a két terminál között:	**621 Mbit/s**
MAC cím vizsgálat esetén "terhelő" gépek között:	**459 Mbit/s**

Átkapcsolási teszt

Az útvonalmódosító bejegyzést minden ötödik másodpercben léptettük érvénybe. Az így kapott eredmények egy 100 Mbit/s sávszélességű UDP adatkapcsolat esetén:

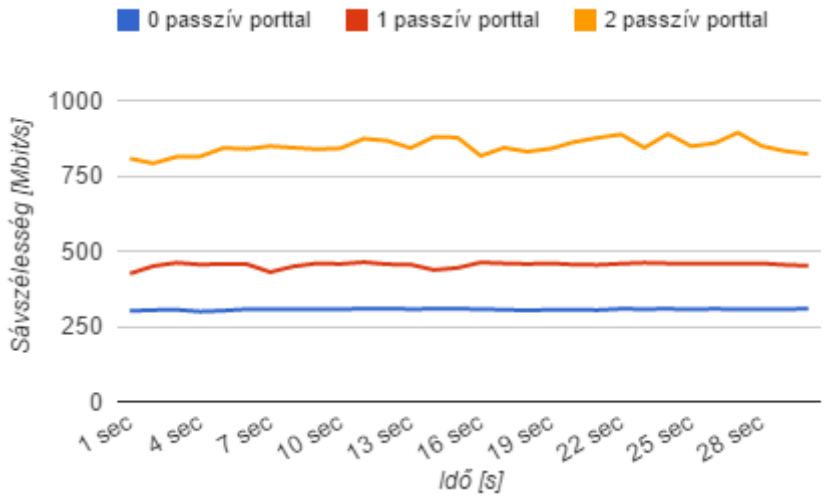
5. táblázat: MikroTik RB2011 - Átkapcsolási veszteségek

	csomagveszteség	**sorrendi hibák**
1. folyammódosítás	3/8547 (0,035%)	0
2. folyammódosítás	3/8547 (0,035%)	0
3. folyammódosítás	4/8526 (0,047%)	1
4. folyammódosítás	1/8548 (0,012%)	0

Minden további időpontban az összes csomag helyes sorrendben érkezett meg és csomagveszteség is 0% volt.

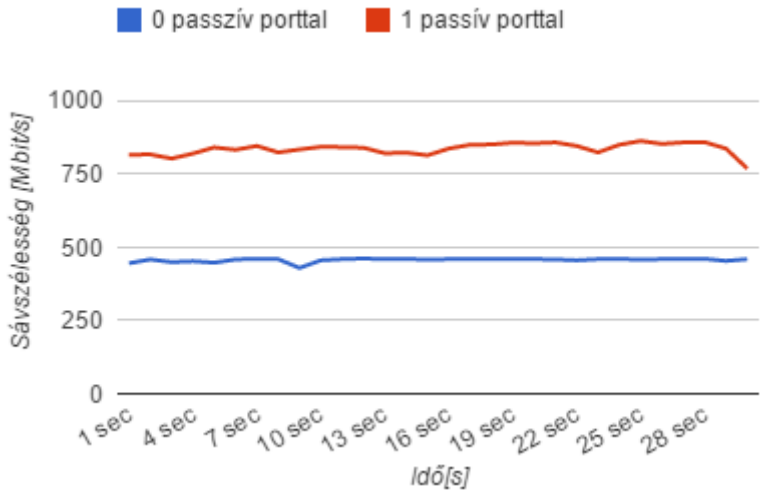
Csakúgy, mint az előző MikroTik-nél, itt is tapasztaltuk az összeomlást 300 Mbit/s sávszélességnél.

Multicast / Broadcast teljesítmény



40. ábra: MikroTik RB2011 - Broadcast teszt

A 40. ábrán látható sávszélességek átlagai **306 Mb/s** (0 passzív port), **454 Mb/s** (1 passzív port) és **847 Mb/s** (2 passzív port) voltak.



41. ábra: MikroTik RB2011 - Multicast teszt

Az átlag sávszélességek az **456 Mb/s** (0 passzív port), **835 Mb/s** (1 passzív port) volt.

UDP protokoll esetén az átlageredmények a következőek lettek:

Broadcast továbbítás passzív port nélkül:	**298 Mbit/s**
Broadcast továbbítás egy passzív porttal:	**445 Mbit/s**
Broadcast továbbítás két passzív porttal:	**801 Mbit/s**
Multicast továbbítás passzív port nélkül:	**430 Mbit/s**
Multicast továbbítás egy passzív porttal:	**795 Mbit/s**

Mindezek után végeztünk az OpenFlow switch-ek tesztelésével, így tiszta képet kaptunk az OpenFlow kompatibilitásukról. Azonban célunk nem csak az adatok rögzítése volt, hanem az egyes eszközök összehasonlítása is. Ezt a következő fejezetben részletezzük.

5. ESZKÖZÖK ÖSSZEHASONLÍTÁSA ÉS ÉRTÉKELÉSE

Az előző fejezetben teszteltük az összes rendelkezésünkre álló eszköz OpenFlow-képességét az általunk definiált méréscsoportok szerint. Ebben a fejezetben a kapott eredmények alapján összehasonlítjuk az eszközöket.

Először összegezzük azokat az ismereteket, melyekkel rendelkeznünk kell a kontroller programozásához a hálózat kiépítésénél, a hibás működés elkerülése érdekében.

6. táblázat: Alapvető OpenFlow működés összehasonlítása

Eszköz	**Támogatott táblák száma**	**Bejegyzések mentése**	**Teli tábla jelzés**	**Hardveres feldolgozás**
HP 1.0	1	OK	Implementálva	Van
HP 1.3	max 5	OK	Implementálva	Van
MikroTik RB750GL	1	Csak kontroller jelenlétében	Nincs implementálva	Nincs
MikroTik RB2011	1	Csak kontroller jelenlétében	Nincs implementálva	Nincs

A táblázatból kiolvasható, hogy a hardveres továbbítás megvalósítása a legnagyobb különbség a switch-ek között. Értelemszerűen a HP switch-ek esetében olyan kontroller-t és hálózatot kell tervezni, mely ezt az adottságot ki tudja használni, így a lehető leggyorsabb és legmegbízhatóbb működést kapjuk meg. Következő lényeges vizsgálnivaló a teli tábla jelzés implementálásának megvalósítása, ugyanis az eszközök memóriagazdálkodása szempontjából ez az információ rendkívül fontos a vezérlő számára egy gyakran változó, sok folyambejegyzéssel működő hálózatban. Összegezve az *Alapvető OpenFlow működési tesztek* kategóriának eredményeit, a HP switch-ek választása esetén megfontoltabb kontroller programozási feladat vár a hálózatépítőre a hardveres kritériumok megfelelése miatt. A MikroTik switch-eknél pedig fontos kalkulálni a teli tábla jelzés hiányával – ami a nagyméretű, dinamikus és sok folyambejegyzésekkel dolgozó hálózatok esetén kritikus.

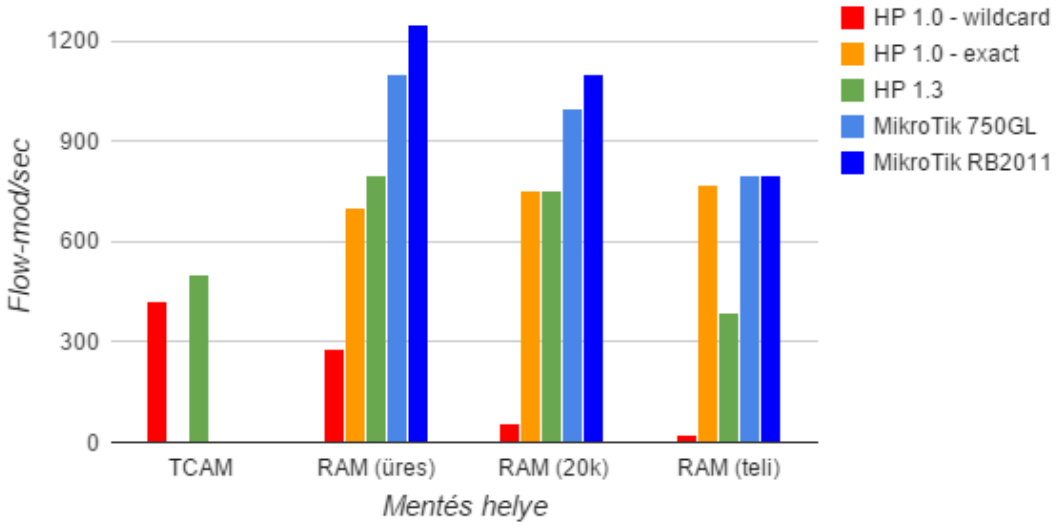
Ezután az *Elemi OpenFlow műveletek teljesítményének* összehasonlításai következnek.

7. táblázat: Táblakapacitások worst-case esetben

Eszköz	**Szoftveres tábla (RAM)**	**Hardveres tábla (TCAM)**
HP 1.0	65.565 - TCAM-ben lévő bej.	max 1500 bejegyzés
HP 1.3	65.565 - TCAM-ben lévő bej.	max 1500 bejegyzés
MikroTik RB750GL	egyszerre 26.390 bejegyzés maximum ~100 ezer bejegyzés	-
MikroTik RB2011	egyszerre 30.387 bejegyzés maximum ~100 ezer bejegyzés	-

Látható, hogy a HP rendelkezik nagyobb táblakapacitással. Azonban megjegyezendő, hogy ha egy hálózat programozó számol a MikroTik-ek esetében tapasztalt bejegyzés feltöltés megszakadásával és úgy programozza a kontroller-t, hogy ez ne jelentsen problémát, akkor közel 108 ezer folyambejegyzés mentésére képesek ezek az olcsó eszközök. Ebben az esetben majdnem kétszeresét képesek elmenteni a HP maximális kapacitásának.

Következő lépésben vizsgáljuk meg az egyes eszközök folyambejegyzéseinek mentésének idejét:



42. ábra: Flow-mod teljesítmények összehasonlítása

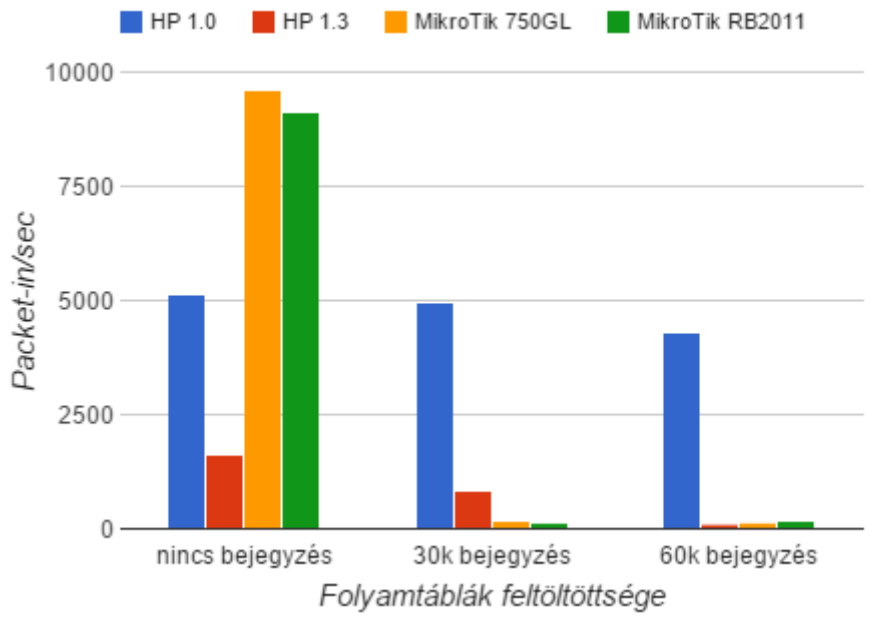
Lényeges különbség a wildcard és az exact bejegyzések közötti feltöltésnél csak a HP 1.0-nál volt tapasztalható, így annak az értékeit szeparáltuk. Látható, hogy a MikroTik

RB2011 switch rendelkezett minden esetben a legnagyobb flow-mod/sec értékkel, vagyis ez a kapcsoló tudja a legkevesebb idő alatt elmenteni a legtöbb folyambejegyzést.
A teli táblák törlésének ideje az alábbi táblázatban látható:

8. táblázat: Törlési idők összehasonlítása

Eszköz	**Teli tábla törlés ideje**
HP 1.0 - wildcard bejegyzésekkel	összeomlott a rendszer
HP 1.0 exact bejegyzésekkel (65k)	2,09 másodperc
HP 1.3 wildcard bejegyzésekkel (65k)	23,49 másodperc
MT 750GL - 65546 db wildcard bejegyzéssel	27,93 másodperc
MT RB2011 - (66894db bejegyzés)	28,56 másodperc

A packet-in/sec eredményekben a következő ábrán láthatóak a különbségek:

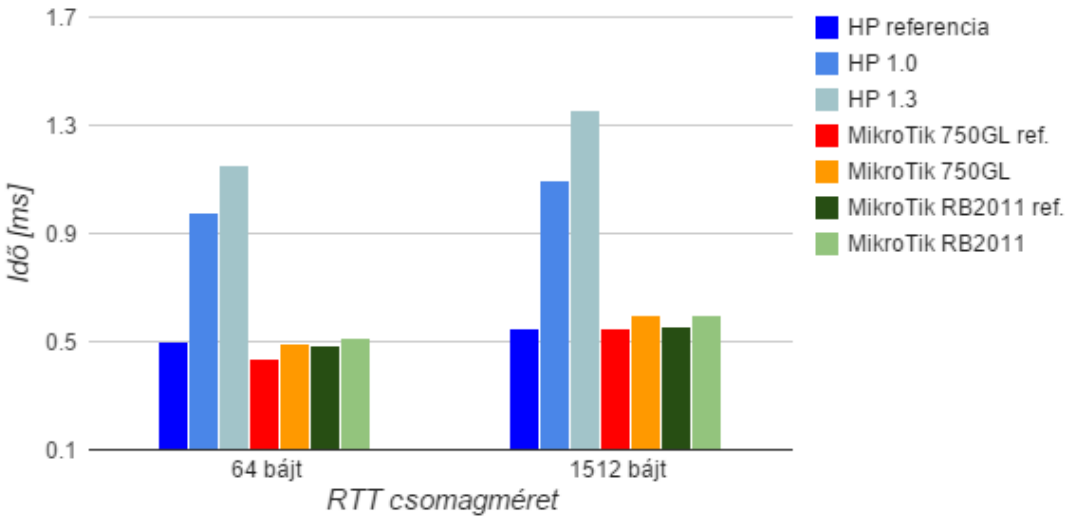


43. ábra: Packet-in teljesítmény összehasonlítás

A HP 1.0 kivételével amint a táblák több folyambejegyzést tartalmaznak a packet-in üzenetek generálása drasztikusan lecsökken. Ezzel a releváns információval feltétlen számolni kell, mivel egy reaktív on-demand hálózati rendszerben – ahol a kontroller a packet-in üzenetek alapján alakítja ki megfelelő továbbítást (pl.: pontos útvonal) a célcím felé – kritikus fontosságú a termékek packet-in/sec teljesítménye.

A továbbiakban tekintsük át az utolsó, *Magasszintű hálózati teljesítménytesztek* eredményeit!

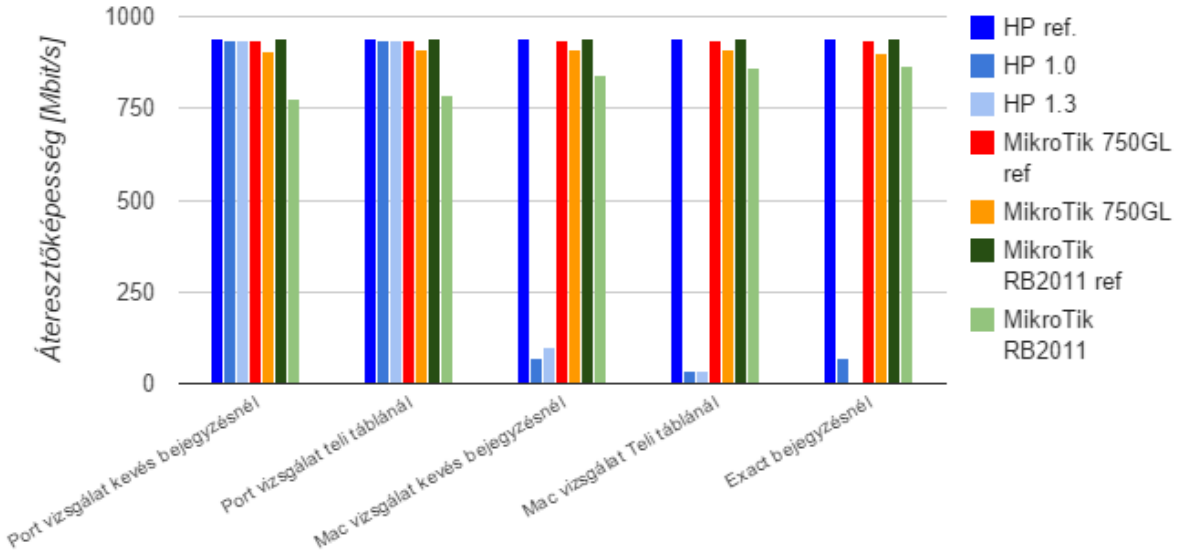
RTT:



44. ábra: RTT idők összehasonlítása

Jól látható, hogy a HP switch esetében igen nagy overhead-et jelent az OpenFlow használat. Ezzel ellentétben a MikroTik eszközöknél minimálisan növekedett csak a körül fordulási idő a hagyományos kapcsoláshoz képest.

Áteresztőképesség worst-case esetben:



45. ábra: Áteresztőképességek összehasonlítása

Minden eszközön megmértük az OpenFlow nélküli teljesítményt, referencia értékként használva. Ezek mindhárom eszköz esetén 940 Mb/s körül alakultak.

A HP esetén OpenFlow feldolgozáskor jól látható, hogy amíg a folyam hardveresen kezelt, természetesen képes a referencia sebességgel (**936 Mb/s**) üzemelni. Viszont drasztikusan csökken a teljesítmény (**34-100 Mb/s**) amint szoftveresen kezelt a feldolgozás.

Ezzel szemben a két MikroTik OpenFlow feldolgozása kedvező körülmények (kevés bejegyzés) között is eleve lassabb (MikroTik 750GL: **906 Mb/s**, MikorTik RB2011 **776 Mb/s**), viszont worst-case esetben (sok bejegyzés) is képes tartani ezt a teljesítményt.

Nagyon fontos észrevétel tehát, hogy amennyiben a HP switch a hardveres feldolgozás kritériumait sértő folyambejegyzésekkel kényszerül dolgozni (vagy kifogy a TCAM memória-területekből), akkor az drasztikusan kisebb teljesítményre képes a nála nagyságrendekkel olcsóbb eszközökhöz képest.

Érdekes kérdés, hogy a nagyobb CPU-val rendelkező MikroTik RB2011 miért ér el rosszabb teljesítményt. A megoldás a CPU + RAM OpenFlow implementálási módban keresendő, ugyanis míg a 750GL esetében a továbbításnál a CPU kihasználtság ~97% volt, a másiknál ez csak ~60% körüli teljesítményt ért el. Mivel tudjuk, hogy a MikroTik eszközök, csak szoftveres folyamfeldolgozást valósítanak meg, így egyértelmű, hogy az alacsonyabb teljesítmény a kisebb processzor kihasználtság miatt adódik.

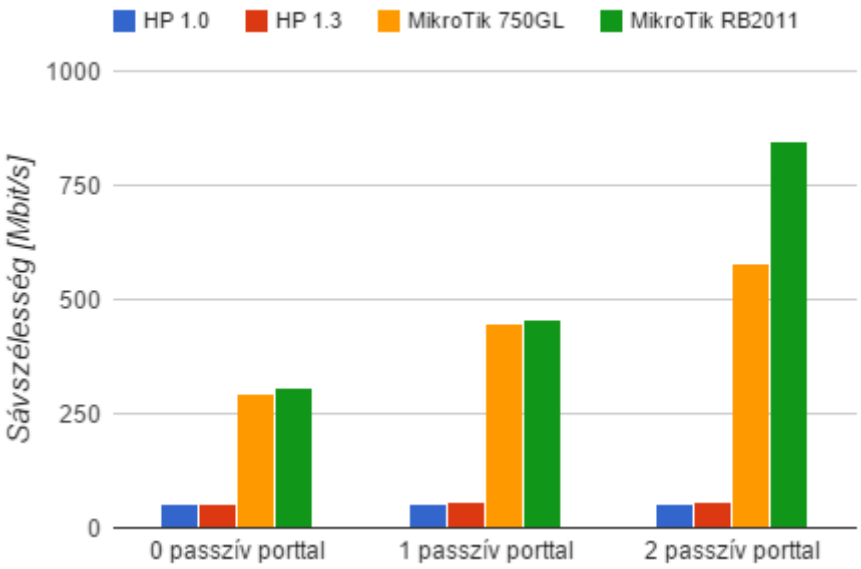
Átkapcsolási teszt

9. táblázat: Átkapcsolási veszteségek összehasonlítása

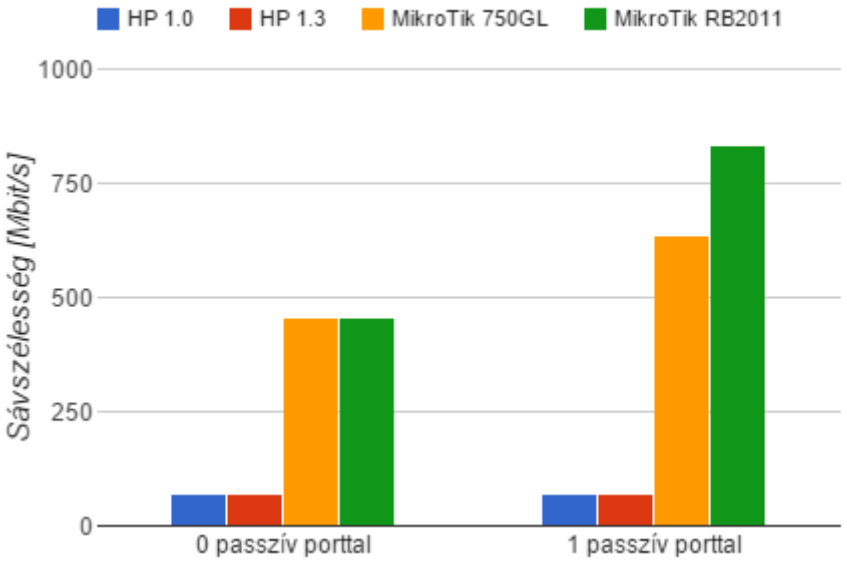
Eszköz	**csomagvesztés**	**sorrendhiba**
HP 1.0	0,14%	190
HP 1.3	0,13%	157
MikroTik 750GL	0,038%	1
MikroTik RB2011	0,032%	1

A teszt eredményeiből azt láthatjuk, hogy a dinamikus útvonal-módosítás mekkora mértékben befolyásolja az adott eszközön éppen működő adatkapcsolatot. Érdekes jelenség, hogy a MikroTik termékek sokkal jobban kezelik a folyambejegyzések módosítását, mint a jelentősen drágább HP switch.

Broadcast/Multicast



46. ábra: Broadcast teljesítmények összehasonlítása



47. ábra: Multicast teljesítmények összehasonlítása

A broadcast-ot és mulitcast-ot használó hálózatok esetében értelemszerűen a HP switch-ek hátrányban vannak a MikroTik-ekkel szemben, hiszen a több portra történő továbbítás szoftveresen kezelt művelet és ezt a HP eleve alacsony teljesítményen tudja csak kiszolgálni. Mivel a MikroTik eszközök gyors szoftveres feldolgozásra képesek, ezen a területen előnyben vannak. Azonban, ahogy több portra kell továbbítaniuk a forgalmat, úgy az egyes portok között egyre jobban eloszlik a maximális átviteli kapacitás. Így is jobb teljesítményt nyújtanak, mint a HP, de erre a megkötésre figyelnünk kell.

Röviden összegezve az eredményeket, a következőket állapítottuk meg a HP és MikroTik-ek összehasonlításakor:

- MikroTik eszközöknél feltétlen szükséges a kontroller helyes programozása, hogy a tapasztalt hibák ne korlátozzák a hálózat működését.
- Ha a HP vonali sebességű továbbítását kívánjuk használni, akkor alaposabb hálózat tervezés és kontroller programozás szükséges, mint a MikroTik esetében.
- Sok folyambejegyzéssel operáló hálózatokban a HP táblakapacitása és megbízhatósága nagyobb, mint a MikroTik-nél (kivéve a kontroller olyan módon történő programozását, ami kizárja a táblafeltöltési hibákat).
- Folyambejegyzések mentése a MikroTik eszközöknél gyorsabban megtörténik, mint a HP-nál.
- Folyambejegyzések törlése a HP switch esetében minimálisan gyorsabb, mint a MikroTik-eknél.
- Packet-in üzenetek küldése nem skálázódik jól a MiktroTik-eknél. A switch-ek packet-in üzeneteire gyorsan reagáló hálózatban jobban megfelel a HP.
- Hardveres feldolgozás esetében a HP áteresztőképessége a legnagyobb, de minden egyéb csomagtovábbításnál alulmarad a MikroTik-kel szemben.
- Folyambejegyzés módosító üzenetek a keresztül haladó adatfolyamot jobban zavarják a HP-nál, mint a MikroTik esetében.
- Multicast és broadcast használatkor a MikroTik biztosít jobb teljesítményt.

Összességében a HP switch egyértelműen jobb választás egy olyan OpenFlow hálózat esetén, ami megfelel a HP hardveres feldolgozás-kritériumainak. Ellenkező esetben viszont a nála nagyságrendekkel olcsóbb MikroTik nyújt jobb teljesítményt.

6. Összefoglalás

Munkánk során felismertük, hogy egy eszköz SDN-re való alkalmasságának vizsgálatánál az SDN *de facto* szabványát, az OpenFlow protokoll implementálásának módját és az így kapott teljesítményét szükséges elemezni. Mivel minden gyártó a legkevesebb költség ráfordításával próbálja implementálni a protokollt, így új hardver tervezése helyett, a már létező, saját architektúrájukra alkalmazva igyekeznek megvalósítani azt. Az így kialakult heterogén piaci környezetben az egyes eszközök működése és teljesítménye nagyon eltérő lehet.

A valódi hardveres OpenFlow implementációk ezen problémája égető kérdés, mégis kevés kutatás foglalkozik az eszközök teljesítményének átfogó vizsgálatával. Habár az ONF-nek létezik OpenFlow megfelelőségi programja, azonban ezek a tesztek csak a szabványban előírt fejlécmezők illeszkedés-vizsgálatának implementálását ellenőrzik. Ezzel ellentétben a munkánk kulcselemenként kidolgozott mérési módszertanunk átfogó képet ad az OpenFlow-t támogató különböző termékek valós környezetben, valós feladatok elvégzésekor nyújtott teljesítményéről.

Ezután a kidolgozott módszertant követve méréseket végeztünk a rendelkezésünkre álló valós hálózati eszközökön, majd az összehasonlításnál megmutattuk, hogy a jelenlegi OpenFlow piaci környezetben a több nagyságrendben eltérő árú eszközök között (akár az eszköz-specifikációknak is ellentmondóan) felborul a használattól függő ár-érték arány.

IRODALOMJEGYZÉK

[1] Open Networking Foundation OpenFow specifikációk, web: https://www.opennetworking.org/sdn-resources/onf-specifications/openflow

[2] OpenFlow Switch Specification, version 1.0.0, 2009. Dec., web (letöltve 2014.08.11): https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf

[3] OpenFlow Switch Specification 1.3.1 (Sept. 6, 2012), web (letöltve 2014.08.11): https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.1.pdf

[4] NFV, Network Virtualization, OpenFlow & SDN Use Cases, web (letöltve 2014.09.23): https://www.sdncentral.com/sdn-use-cases/

[5] H. Yamanaka, E. Kawai, S.i Ishii és S. Shimojo: OpenFlow Networks with Limited L2 Functionality, web (letöltve 2014.10.12): http://www.thinkmind.org/download.php?articleid=icn_2014_9_40_30142

[6] J. Liao, "SDN system performance," 2012, web (letöltve 2014.09.23): http://pica8.org/blogs/?p=201

[7] K. Pagiamtzis és A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," IEEE Journal of Solid-State Circuits, vol. 41, no. 3, 2006.

[8] P. Bosshart, G. Gibb, H.-S. Kim, G. Varghese, N. McKeown, M. Izzard, F. Mujica, M. Horowitz: Forwarding Metamorphosis: Fast Programmable Match-Action Processing in Hardware for SDN, web (letöltve 2014.09.15): http://yuba.stanford.edu/~grg/docs/sdn-chip-sigcomm-2013.pdf

[9] A. Greenberg: VL2: A Scalable and Flexible Data Center Network. SIGCOMM, Aug. 2009.

[10] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, S. Banerjee: DevoFlow: Scaling Flow Management for High-Performance Networks, web (letöltve 2014.09.27): http://conferences.sigcomm.org/sigcomm/2011/papers/sigcomm/p254.pdf

[11] S. Kandula, S. Sengupta, A. Greenberg, P. Patel: The Nature of Datacenter Traffic: Measurements & Analysis. IMC, 2009.

[12] T. Benson, A. Akella, D. A. Maltz: Network Traffic Characteristics of Data Centers in the Wild, web (letöltve 2014.10.07): http://conferences.sigcomm.org/imc/2010/papers/p267.pdf

[13] Open vSwitch ovs-ofctl manual, web (letöltve: 2014.08.19): http://openvswitch.org/cgi-bin/ovsman.cgi?page=utilities%2Fovs-ofctl.8

[14] WireShark letöltő oldal, web: https://www.wireshark.org/download.html

[15] Nping fejlesztői oldal, web: http://nmap.org/nping/