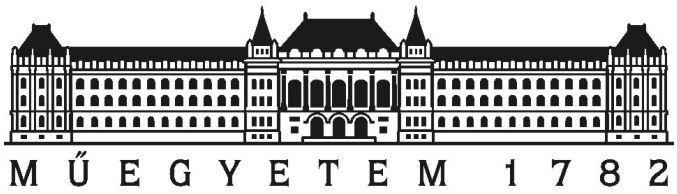


Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Hálózati Rendszerek és Szolgáltatások Tanszék

Szolgáltatásromlás észlelése a hálózati forgalom korai áramlási jellemzőinek felhasználásával

Makara László Árpád
G5YPX8

Konzulens:
Dr. Pekár Adrián

2022. november 1.

Kivonat

A digitalizáció következtében egyre több eszköz és szolgáltatás kapcsolódik az internet világához, ebből kifolyólag a végeszközök és az általuk generált forgalom mérete is exponenciálisan megnövekedett. A felhasználói elégedettség a szolgáltatók nézetéből kritikus támpont, így a felhasználók által támasztott elvárásokat minél jobban garantálnia kell az egyes szolgáltatóknak. Ennek elengedhetetlen kelléke a hálózat valós idejű monitorizálása és a szolgáltatásromlás felismerése. A minőségjavítási és megőrzési lépéseket viszont a szolgáltatók tulajdonában lévő, a felhasználók használatára bocsájtott hálózati eszközök minimális költségtöbleti vonzatával, illetve a már meglévő hardverkészlet cseréje nélkül célszerű elérni az optimalizálási és szavatolási eljárások alkalmazása mellett. Egy a korai áramlási jellemzőkre alapozó megoldás alkalmasnak tűnik az ilyen hálózati eszközök szolgáltatásminőség konfigurációjának dinamikus finomhangolására a forgalmi viszonyok függvényében. Ezen dinamikus konfigurációmódosítások az egyes adatfolyamok kölcsönös koherenciájára alapoz, amely függvényében prediktív modellek alkalmazhatók.

Ezen dolgozat a korai áramlási jellemzők felhasználási lehetőségeit vizsgálja a szolgáltatásromlás észlelése érdekében, amely a hardveres gyorsítótárba kerülést megelőzvén, a hálózati forgalom kezdeti stádiuma szoftveres lekövethetőségének feltevésből származtatódik. Javaslatot tesz az optimális paraméter konfigurációkra, amely biztosítja a szolgáltatásminőség fenntartását az erőforrásigény minimalizálása figyelembe vételével, illetve a szolgáltatók eszközkészletének cseréje elkerülésével. Továbbá taglalja a gyakorlati alkalmazási lehetőségeket és feltárja az egyes adatfolyamok közötti kohéziókat, amelyek által az egyes implikációk beláthatókká válnak a szolgáltatásromlás terén. Ezen implikációk feltárásával lehetőség nyílik a problémák absztrahálására és memóriahatékony eljárások célzott alkalmazásának kidolgozására.

Abstract

As a result of digitalization, more and more devices and services are connected to the Internet, and the size of the end devices and the traffic they generate has increased exponentially. User satisfaction is a critical benchmark from the perspective of service providers, so they need to guarantee their users' expectations as much as possible. Real-time network monitoring and service degradation detection are essential for this. On the other hand, quality improvement and preservation measures should be achieved with minimum cost overheads for the network assets owned by the service providers and made available to users without replacing existing hardware while applying optimization and guarantee procedures. A solution based on early flow characteristics seems suitable for dynamic fine-tuning of the quality of service configuration of such network devices according to traffic conditions. These dynamic configuration adjustments are based on the mutual coherence of the individual data flows, depending on which predictive models can be applied.

This thesis explores the potential of using early flow characteristics to detect service degradation, derived from the assumption of software traceability of the initial stage of network traffic prior to hardware caching. It proposes optimal parameter configurations that ensure the maintenance of service quality by considering the minimization of resource requirements and avoiding the replacement of the service providers' equipment pool. Furthermore, it articulates practical application options and explores the coherencies between data streams, whereby the implications of each can be seen in terms of service degradation. By exploring these implications, it is possible to abstract away the problems and develop targeted applications of memory-efficient techniques.

Köszönetnyilvánítás

Ezúton szeretném megköszönni témavezetőmnek, Dr. Pekár Adriánnak a kutatásom során nyújtotta idejét, támogatását, illetve szakmai és konstruktív megjegyzéseit.

Különleges köszönet illeti továbbá Tuszit és Danit. A munkámhoz nyújtott segítségetek nélkülözhetetlen volt.

Nem utolsó sorban, köszönet illeti családom többi tagját is az évek során nyújtotta támogatásukért és a kiegyensúlyozott családi háttér biztosításáért. Köszönöm, hogy átsegítettetek az igazán nehéz időszakokon!

Tartalomjegyzék

Ábrák jegyzéke

Táblázatok jegyzéke

Rövidítések jegyzéke

BF Bidirectional Flow

biFlows bidirenctional Flows

CDF Cumulative Distribution Function

CPU Central Processing Unit

DstIP Destination IP

DstPort Destination Port

eBPF Extended Berkeley Packet Filter

fEnd flow End

FIN Finish

FPGA Field-Programmable Gate Array

fStart flow Start

HH Heavy Hitter

HOPOPT IPv6 Hop-by-Hop Options

ICMP Internet Control Message Protocol

IP Internet Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

IPv6-ICMP Internet Control Message Protocol for IPv6

ISP Internet Service Provider

k-nn k-neareset neighbor

KPI Key Performance Indicators

LAN Local Area Network

MOGA Multi-Objective Genetic Algorithm

MTU Maximum Transmission Unit

PCAP Packet Capture

PDF Probability Distribution Function

PIAT Packet-Inter-Arrival-Time

PktCount Packet Count

PS Packet Size

QoE Quality of Experience

QoS Quality of Service

RAM Random Access Memory

RST Reset

SLA Service Level Agreement

SrcIP Source IP

SrcPort Source Port

SSL Secure Sockets Layer

TCP Transmission Control Protocol

UDP User Datagram Protocol

UF Unidirectional Flow

uniFlows unidirectional Flows

WAN Wide Area Network

Bevezetés

A modern ISP szintű hálózati útválasztok esetében gyakran alkalmaznak adatfolyam alapú hardveres gyorsítást. Az első beérkezett n csomag után a feldolgozása ezeknek az adatfolyamoknak az FPGA szintjén történik meg. Ennek köszönhetően az útválasztok feldolgozási sebessége meghaladhatja az 1 Gbps sebességet a CPU erőforrás használat minimalizálásával. Mindezekből következik, hogy alacsonyabb költségek mellett lehet üzemeltetni nagyobb sebességel a hálózatot. A költségminimalizáció megjelenik a hardver erőforrás méretezése és az energiafelhasználásában egyaránt. Viszont mindennek megvan a saját hátülütője ami ebben az esetben a hálózati monitorozás drasztikus korlátozása, hiszen az n korlátényező adatfolyamonként teljes mértékben behatárolja a szoftveres előrejelzési és monitorozási lehetőségeket.

Az ISP-k szemszögéből kitüntetett figyelemmel bír a monitorozás és a hálózati degradáció felismerése az egyes adatfolyamokban. Munkámban igyekszem egy olyan megoldást találni, amely a korlátozó tényezők figyelembevételével effektíven tud előrejelzést tenni a hálózati degradációban úgy, hogy az energiahatékonyság és a sebesség nem kerül feláldozásra a cél érdekében. A hálózati degradáció miatt szolgáltatásromlás következik be a végfelhasználónál, így az előírt SLA-k nem teljesíthetőek. A degradációt már korai fázisában fel kellene ismerni ahhoz, hogy biztosítani lehessen a szolgáltatás minőségét, miközben a definiált sebesség kárára nem mehet.

Az n korlátényező nem más mint egy adotfolyam első n. darab üzenetváltása a két kommunikációs fél között. A kommunikációs tér két részre bontható ebből a szemszögből, hiszen az első n darab csomag lesz a megfigyelt térbe elhelyezkedőek, míg a folyamból hátralévő elemek a nem-megfigyelt térben helyezkednek el. A nem-megfigyelt csomagok hiányosságát megfogni nehézkes, mivel nem ismerünk egy egzakt számot az adott alkalmazás típusok esetén, hogy hány csomagból fog az adott adatfolyam állni. Ezeken felül fontos megkülönböztetni a kommunikáció irányultságát is. Az egyik a WAN oldali késleltetés a másik pedig a LAN oldali. A kettő elszeparálása fontos, hiszen a szolgáltatói perspektívából csakis a LAN oldali késleltetését tudja befolyásolni, hiszen a WAN oldalon megjelenő szerver késleltetés vagy szolgáltatás kiesést nem tudja befolyásolni.

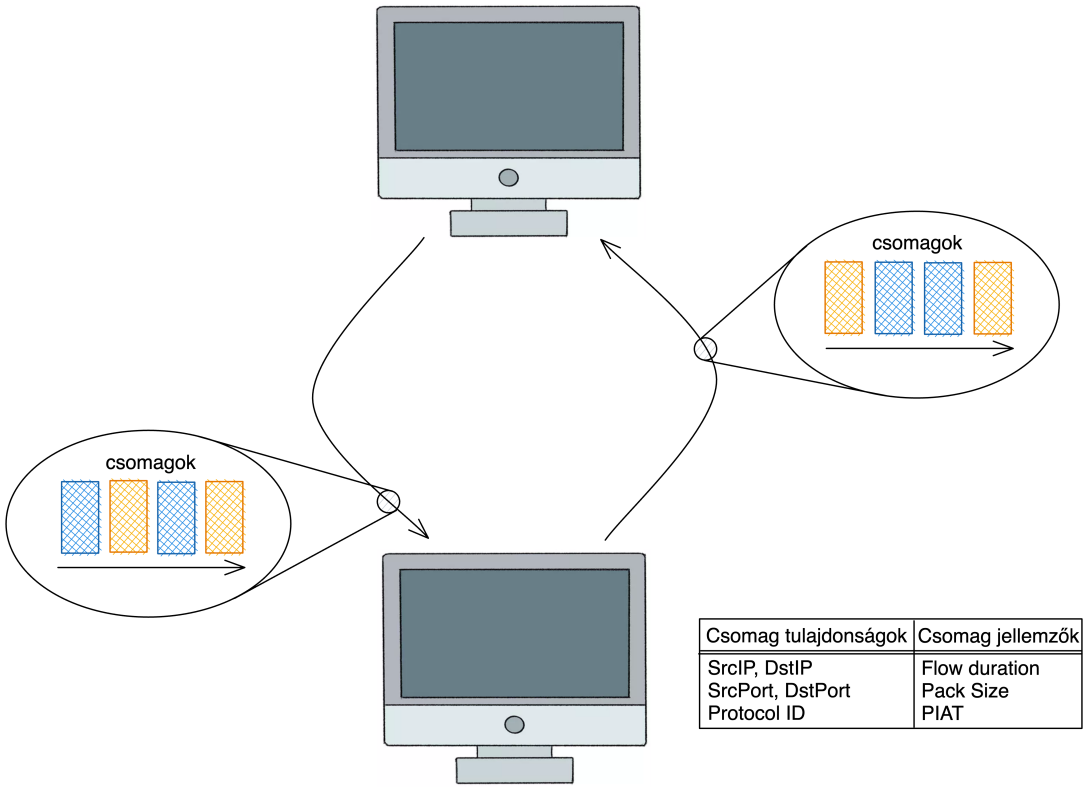
Munkám során a következő kérdésekre kívánok választ találni és javaslatot tenni az optimális csomag vizsgálati méretre: *(*i) Az n megfigyelt csomagok függvényében mekkora százalékos arányban következik be szolgáltatásromlás az áramlás nem-megfigyelhető állapotában. *(ii)* Adott áramlás nem-megfigyelt állapotában észlelt szolgáltatásromlás befolyásolja-e más adatfolyamok megfigyelhető státuszban lévő szolgáltatás állapotát.

A dolgozat további része a következőképpen szerveződik. Az 1. fejezet rövid elméleti hátteret nyújt, beleértve hálózati monitorozást, a csomagfeldolgozási fázisokat, kommunikáció irányultságát és az útvonalválasztók felépítését, valamint tárgyalja a releváns kapcsolódó munkákat. A 2. fejezet leírja a meghatározott korai áramlási jellemzőkkel kapcsolatos mutatókat, a kommunikációs tér javasolt felbontását, valamint a felhasználási eseteket. A 3. fejezet bemutatja a korábban felvázolt használati esetek validálását, beleértve a felhasznált adathalmazt és a megfigyelt analitikus eredményeket. Dolgozatom végén összegzem az elért eredményeimet és megadom a jövőbeli kutatásom tervezett irányát.

1. fejezet

Kutatási háttér és kapcsolódó munkák

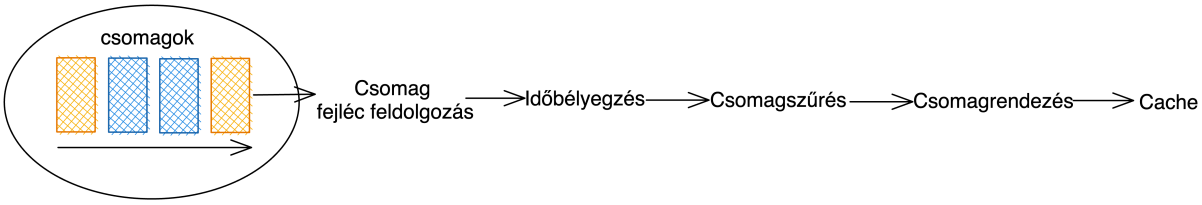
A technológia fejlődés és a felhasználók igényeinek kielégítése függvényében (beleértve az okosotthon, illetve az összes vezetéknélküli hálózatra kapcsolódó okoseszközt) az átlagas otthoni hálózatok komplexitása jelentősen megnövekedett. Egy ISP hálózati topológiáját tekintve a felhasználói QoE és QoS szemszögéből a szűkkeresztmetszet a felhasználói útvonalválasztó végeszköznél jelentkezik. Ahhoz, hogy a lehető legjobb szolgáltatást tudja az ISP garantálni előfizetőinek elengedhetetlen a hálózat lokális monitorozása és az adatfolyamok alapján következtetések megállapítása a dinamikus QoS érdekében. Számos korábbi publikáció foglalkozik a hálózati forgalom monitorozásával [1], [2], viszont ezek valós visszacsatolást önállóan nem biztosítanak a hálózati optimalizációhoz.



1.1. ábra. Áramlási tulajdonságok és áramlási jellemzők közötti kapcsolat.

Az adatfolyam egy megfigyelési ponton áthaladó csomagok halmaza egy adott időtartam alatt. Az azonos folyamhoz tartozó csomagok közös attribútumait 5-tuple-ök határozzák meg. A folyamat során létrehozott 5-tuple általában forrás- és cél IP-címekből, protokollazonosítókból, valamint forrás- és célportokból áll[3]. Az áramlási jellemzők kiszámítása mindig az adott folyamhoz tartozó csomagokon alapul. Megfigyelhető jellemzők például az egy folyamhoz tartozó csomagok mérete (PS), vagy az eltelt idő (PIAT) figyelése az azonos folyamhoz tartozó két egymást követő csomag között. Az áramlás, az áramlási tulajdonságok és az áramlási jellemzők közötti kapcsolatot a 1.1. ábra ábrázolja.

A 1.2. ábrán látható a folyamatábrája a csomagok rögzítésétől az adatfolyam bejegyzések előállításáig bezárólag. A csomagok begyűjtése általában szoftveres monitorizáló eszközökkel történik, de léteznek alternatív hardvereszközök mint az eBPF, amely passzív eszközként képes a hálózati forgalmat futási időben monitorizálni. Mielőtt az egyes adatcsomagok áramlási folyamba rendeződnének először időbélyegzést kapnak, majd egy csomagkiválsztási procedúrán mennek keresztül, ez egy opcionális lépés melyet az erőforrás-korlátok esetén esedékes alkalmazni, amely lehet akár tárhely vagy a hardver erőforrás mint a CPU, RAM hiány. Ennek célja a csomagok számának csökkentése, amely vagy csomagmintavétellel (az összes csomagból csak egy részhalmaz kiválasztása) vagy csomagszűréssel (adott végpontok közötti vagy adott IP tartományon belüliek kiválasztása) történik. A csomagkiválasztási eljárások során a nem szűrt részhalmaz eldobásra kerül. Legutolsó lépés a csomagok áramlási bejegyzésekbe való rendezése. Ezek az áramlási rekordok hordozzák az osztályozáshoz használt áramlási tulajdonságokat.

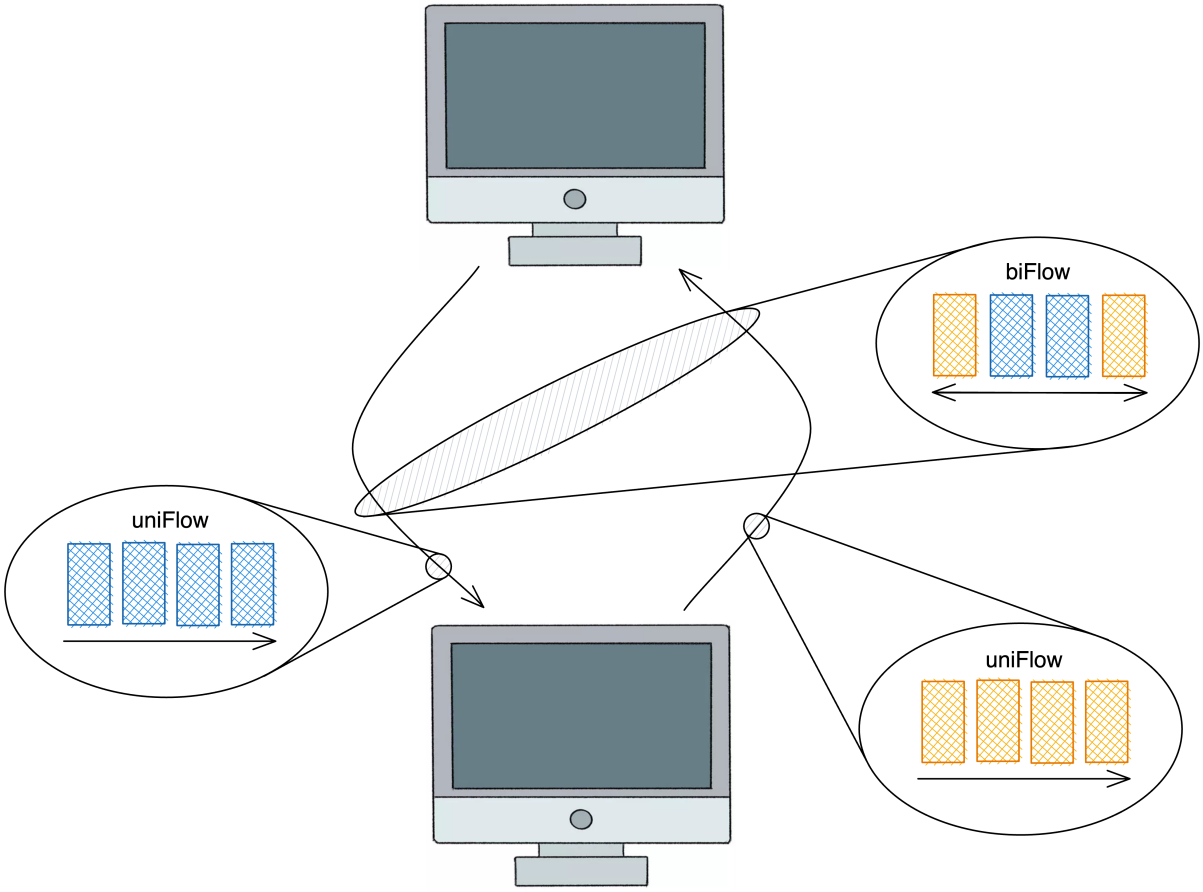


1.2. ábra. Csomagfeldolgozási fázisok a rögzítéstől az áramlási bejegyzésekig.

Az egyes csomagok adatfolyamba való rendezése során a legfontosabb tényező az irányultság meghatározása. Az adatfolyamok két végpont között átvitt csomagok folyamaként írható le.

A csak egyetlen végpontról a másik végpontra küldött csomagokból álló adatfolyamokat uniFlows-nak nevezik. A csomagok által a továbbítás során megtett hálózati útvonalak asszimetrikus jellege miatt azonban az áramlási jellemzők különböző irányokban eltérő statisztikai tulajdonságokat muthatnak. Az irányultság azonosítása ezért létfontosságú szerepet játszik a forgalom mérésében és osztályozásában. A biFlows a mindkét irányban küldött csomagokból álló adatfolyamok gyűjtőneve. Az uniFlows és a biFlows közötti kapcsolatot a 1.3. ábra ábrázolja.

A monitorozás működése során az áramlási bejegyzések egy cacheben vannak tárolva ideiglenesen. Az áramlási bejegyzések karbantartásával kapcsolatos feladatok közé tartozik azok létrehozása, frissítése és lejárat esetén ürítése. A 1.4. ábrán látható az egyes rögzített cso-

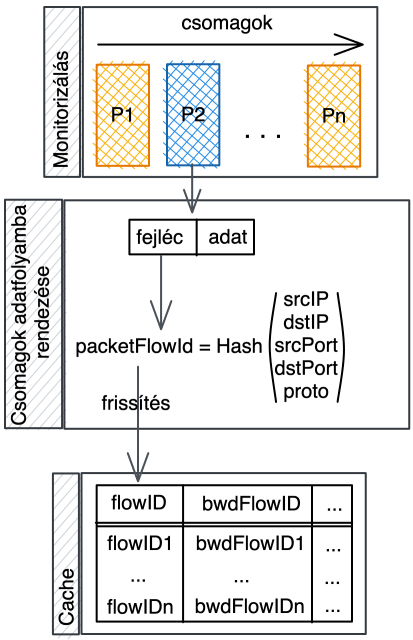


1.3. ábra. Az egyirányú és a kétirányú áramlások közötti különbség.

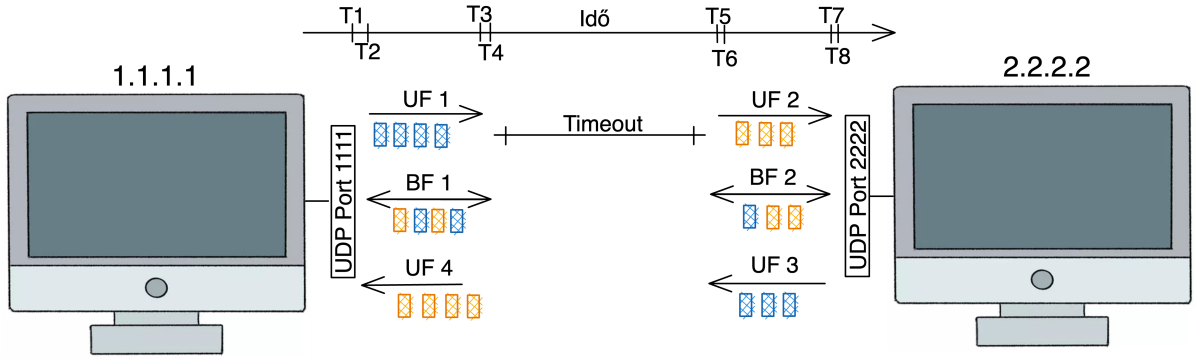
magok elemzésének és összehasonlításának folyamata az áramlási gyorsítótár felhasználásával. Új bejegyzés akkor jön létre, ha egy új csomag nem tartozik a már cache-ben lévő adatfolyamok egyikébe sem. Ez a művelet erőforrásigényes feladat különösen akkor, ha több százezer rekordot kell egyidejűleg fenntartani a gyorsítótárban, ezért az adatfolyamok közötti keresés gyorsítása érdekében a folyamok flowID kulcs bevezetésével történnek. Ez a flowID egy hashelt érték, amelyet az áramlási tulajdonságok alapján számít ki a rendszer (5-tuple). Minden új rögzítendő csomaghoz egy packetFlowID hash érték számítás történik, majd az összevetésre kerül a gyorsítótárban meglévő áramlási azonosítókkal. Találat esetén adatfolyamhoz történő hozzárendelés zajlik le, míg ellenkező esetben pedig egy új adatfolyam létrehozása esedékes. A flowID-k mellett párhuzamosan minden egyes folyambejegyzéshez egy bwdFlowID is tárolásra kerül. Ez a felcserélt forrás és cél IP cím, illetve a portszámok megcserélése után számított hash értékből tevődik ki, amelynek relevanciája a visszafelé irányuló csomagokhoz való illesztés esetén mutatkozik meg.

A 1.5. ábrán látható egy a [1.1.1.1] és a [2.2.2.2] IP című hoszt közötti kétirányú UDP forgalmon végrehajtott adatfolyam összevonási folyamatotot. A származtatott kétirányú áramlási jellemzők egy részhalmaza a 1.1. táblázat írja le.

Az ehhez tartozó áramlási gyorsítótárra látható példa a 1.2. táblázatban. A táblázat egyes sorai az áramlási tulajdonságok által meghatározott áramlási rekordokat tartalmazzák.



1.4. ábra. Csomagok adatfolyamokba rendezése.



1.5. ábra. Csomagok összerendelése biFlow áramlásokban.

A megfigyelt csomagok tulajdonságaitól függően vagy új áramlási rekordok jönnek létre, vagy a meglévő áramlási jellemezők frissülnek. A 1.3. táblázatban látható az új áramlási bejegyzések által okott változások.

Az adatfolyamoknak asszociált gyorsítótárban lévő adatfolyam bejegyzések fenntartásához további feladat a folyamatban lévő áramlások nyomonkövetése és azok lejáratának észlelése. Több oka is lehet az adatfolyamok lejáratának, ha bizonyos ideig nem figyelhető meg az adatfolyamhoz tartozó csomag (idle timeout), vagy pedig az áramlásnak passzív lejárata van. Az adatfolyamok rendszeresen lejárnak, még akkor is, ha az adatfolyamhoz tartozó csomagok áramlása folyamatos (a TCP sajátosságának köszönhetően). Ezt a lejárati időt leggyakrabban active time-nak nevezik. Egy áramlás akkor tekinthető természetes módon lejártnak, ha egy TCP csomagban a FIN vagy az RST jelző megjelenik [4].

A hálózati adatfolyamok korai áramlási jellemzőinek felhasználásával döntő fontosságú előrelépéseket lehetne tenni a szolgáltatok szolgáltatás minőségének javítására. Ismerve a külön-

1.1. táblázat. A biFlows áramlások jellemzőinek egy részhalmaza.

Áramlási tulajdonságok és jellemzők	**biFlow 1**	**biFlow 2**
Id	1	2
Forrás IP cím	1.1.1.1	2.2.2.2
Cél IP cím	2.2.2.2	1.1.1.1
Forrás port	1111	2222
Cél port	2222	1111
Szállítási protokol azonosító	17	17
Forrásból a célállomásra csomagok száma	4	4
Célállomás és forrás közötti csomagok száma	3	4
Áramlás csomagszáma	7	8
Áramlás kezdete	T1	T5
Áramlás vége	T4	T8
Forrás és célállomás közötti időablak	(T4 - T1)	(T7 - T5)
Célállomástól a forrásig Időablak	(T3 - T2)	(T8 - T6)
Áramlás Teljes időtartama	(T4 - T1)	(T8 - T5)

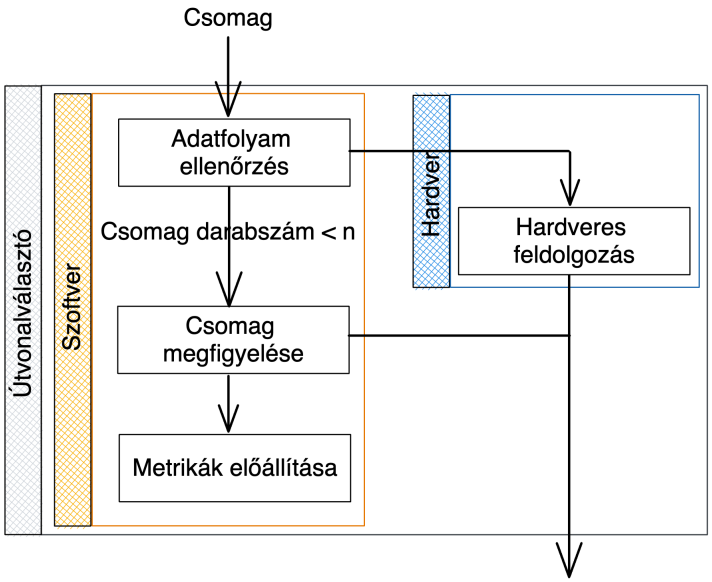
1.2. táblázat. Új áramlási rekordokat tartalmazó cache.

Flow Properties					Flow Features							
SrcIP	SrcPort	DstIP	DstPort	Prot			flowID	PktCount	fStart	fEnd		
147.232.1.1	64100	147.232.2.1	80	TCP	Ftr1	...	#1	5	1002	1035	...	FtrN
147.232.1.2	64123	147.232.2.1	80	TCP			#2	4	1011	1055		
147.232.1.2	64124	147.232.2.3	69	UDP			#3	7	1070	1092		
147.232.1.3	64123	147.232.2.1	443	TCP			#4	10	1347	1805		
147.232.1.3	64124	147.232.2.3	69	UDP			#5	5	1555	1775		

1.3. táblázat. Frissített áramlási rekordokat tartalmazó cache.

Flow Properties					Flow Features								
SrcIP	SrcPrt	DstIP	DstPrt	Prot			flowID	PktCount	fStart	fEnd			
147.232.1.1	64100	147.232.2.1	80	TCP	Ftr1	...	#1	5	1002	1035	...	FtrN	–
147.232.1.2	64123	147.232.2.1	80	TCP			#2	**5**	1011	**1543**			update
147.232.1.2	64124	147.232.2.3	69	UDP			#3	7	1070	1092			–
147.232.1.3	64123	147.232.2.1	443	TCP			#4	10	1347	1805			–
147.232.1.3	64124	147.232.2.3	69	UDP			#5	**6**	1555	**1885**			update
147.232.3.2	64104	147.232.2.3	80	TCP			#6	**1**	**1600**	**1600**			create
147.232.5.3	64111	147.232.2.3	80	TCP			#7	**1**	**1601**	**1601**			create

böző szolgáltatások alkalamzáskategóriáját akár alkalmazás alapú QoS vezérlést lehetne kialakítani a felhasznált szolgáltatásromlás előrejelzés függvényében [5]. A dolgozat egy hálózati forgalomirányító felbontásának feltételezésére épít, ahol a hardveres és szoftveres rész oly módon tevődik össze, hogy a beérkezett adatfolyamok először a szoftver oldalon jelennek meg a fizikai eszközön. A szoftveres oldalon elsőként a kernel megállapítja, hogy hardveres gyorsítás alkalmazása szükséges-e az adott csomagon, melyhez a rendszer tárolja, ismeri a megadott csomag adatfolyamhoz való vonatkozását. Miután azonosította az igényt és nem szükséges még a hardveres gyorsítótárazás akkor az adatfolyam megfigyelési állapotba lép. A megadott csomagszám után az adatfolyam átkerül a hardver oldali lekezelésre és ezek után automatikusan a hardveres gyorsítótárba kerülnek az adatfolyam csomagjai. A leírt folyamat sematikusan a 1.6. ábrán látható.



1.6. ábra. Útvonalválasztó felépítésének referencia modellje.

A felhasználói térben megjelenő metrikák előállítására lépések tudnak szolgálni az ISP-k QoS továbbfejlesztésekre. Aminek köszönhetően jobb minőségben tudnak szolgáltatást biztosítani a végfelhasználók számára.

1.1. Hálózati adatfolyam áramlás mérése

A hálózati adatfolyamok méréséhez és vizsgálatához a nyílt forráskodú NFStream [6] keretrendszert használtam, amely biztosította a magas hálózati átvitel melletti forgalommérést. A keretrendszer felhasználásával biztosított az egységes reprodibilis felhasználási lehetősége a munkámnak más kutatók számára. Számos különböző áramlási jellemzőt rögzít a keretrendszer mint az IP, TCP és UDP csamagcímek, de ezen túlmutatóan alkalmas a korai áramlási jellemzők gyűjtésére is. Nagyszámú számítási elemzés-központú módszert foglal magába a keretrendszer a hagyományos minimum, maximum, átlag és sztandard szóráson túl, melyek alkalmasak a mélyreható analitikák előállítására. Ezeken felül a megoldás lehetőséget biztosít a felhasználó számára, hogy addicionálisan moduláris építkezés mentén kiegészítőket telepítsen a hálózati monitorozáshoz, ezzel saját logikát építve a megfigyelésbe aminek köszönhetően egyedi eljárások építhetőek új hálózati adatfolyam elemző szoftver kreálása nélkül. A létrehozott modulok hordozhatóak és nem igényelnek hardver konfiguráció előírást az NFStream robosztusságának köszönhetően reprodukálható.

1.2. Kapcsolódó munkák

Számos korábbi tudományos munka látható a korai áramlási jellemzők elemzésével kapcsolatban az egyes adatfolyamokra nézve, viszont ezek általában két megközelítési kategóriába

sorolhatóak. Jelentős mennyiségű munka foglalkozik valós időben történő első csomag áramlási jellemzőinek alapján történő kategorizálással [7]–[10]. A másik szemlélet tekintében pedig amikor utólagos feldolgozás keretén belül vizsgálják a korai áramlási jellemzőket, tehát utófeldolgozás történik a hálózati monitorizált adatfolyamokon egy teljesen más aspektusból közelíti meg a problémát [11]–[17].

Bernaille és tsai. [7] a hálózati kommunikáció során közlekedő titkosított csomagok (SSL) osztályozása esetén 85%-os pontosságot értek el a forgalom korai szakaszának felhasználásával. Ezen megközelítés hiányosságára mutattak rá Bar - Yanai és tsai. [8], majd egy tanuláselméleti és statisztikai alapokra épülő osztályozási technikát javasoltak a titkosított hálózati forgalom alkalamzás kategóriákba való sorolására. A megoldásuk a k-nn és k-közép algoritmusok kombinációjából tevődik össze és valós idejű beágyazott környezetbe implementálták és integrálták az eljárást. A kírésleti eredmények alapján a megoldásuk robusztus és effektívebb eredményeket produkált versenytársaikhoz képest.

Dainotti és tsai. [9] rámutattak, hogy az osztályozás oldalán történő lépések hiányosságokat tartalmaznak. Ennek kiküszöbölése érdekében egy automatikus kombinációs technikát mutattak be. Munkájuk során egyaránt alkalmaznak hagyományos és új megközelítési forgalmi osztályozási technikákat, ahol kombinálják az áramlások statisztikai tulajdonságait a csomagok hasznos adatából kinyert információkkal. Rálátást biztosítottak, hogy a kiegészítő osztályozók kiválasztásakor az egyes kombinációs algoritmusok további javulást tesznek lehetővé a már korábban alkalmazott osztályozási technikákhoz képest. Ezen információk tudatában igyekezett Kumano és tsai. [10] egy valós idejű alkalmazás azonosító eljárást megvalósítani titkosított hálózati forgalmon. Az eljárás során rámutattak a titkosított adatok esetén mekkora korai áramlási jellemzőt kell megfigyelni a lehető legnagyobb pontosság elérése érdekében.

Korábbi kutatások során már rálátást biztosított Wright és tsai. [11] a titkosított hálózati forgalom alkalmazás osztályozási lehetőségeire a protokoll információk felhasználásával. Ezen kutatás során megmutatták, hogy kevesebb információ elegendő az alkalmazás kategória sikeres megállapítására a titkosított hálózati forgalmon. Az osztályozó eljárás 90%-nál nagyobb pontosságot ért el mindamelett, hogy a legtöbb protokoll esetén 80%-nál nagyobb pontossággal bírt mélyrehatóbb többosztályos kategórizálásban. Ugyanezen a megközelítési szemszögből igyekeztett Bacquet és tsai. [12] genetikus algoritmust adni az alkalmazás klaszterezésre. A klaszterek számosságát áramlás alapú reprezentációs számításból állapították meg, amely nem használja fel a portszámokat, IP címeket, illetve a hasznos terhet sem. Az így előállt modell 90%-os felismerési arányt biztosít a 14 támogatott alkalmazás osztályra nézve. Okada és tsai. [15] már korábban tettek kísérletet a forgalom jellemzőinek titkosítása miatti változásai alapján történtő módszerek pontosítására. A kírésleti eredmények alapján az eljárás során 28,5%-kal javították az alkalmazás azonosítási pontosságot. Ezen túlmenően az áramlási jellemzők legjobb kombinációját használó azonosítási eljárás nagy pontosságot tett lehetővé kevesebb számítási igény mellett.

A gépi tanulás széleskörű elterjedésének köszönhetően kihatással volt a hálózati forgalmi osztályozásra is. Alshammari és tsai. [13] rámutattak, hogy gépi tanuláson alapuló forga-

lomosztályozás sokkal jobban tud teljesíteni a hagyományos megközelítésű algoritmusokhoz képest. Öt különböző mesterséges intelligencia eljárást alkalmazva rálátást nyújtottak a gépi tanulás alkalmazhatóságára a hálózati osztályozási feladatok során. Ebből inspirálódva Arndt és tsai. [14] három különböző gépi tanuláson alapuló eljárást vizsgálta meg a titkosított hálózati forgalmon történő alkalmazhatóságra. Megmutatták, hogy a tanításhoz használt adathalmaz esetén a folyamatos mintavételezésből származó adathalmaz nem jobb mint az időben véletlenűl mintavételezett hálózati forgalomból származó információ, valamint a MOGA alapú modell óriasi mértékben csökkenti a k-közép alapú klaszterező algoritmusok bonyolultságát. Később Bacquet és tsai. [16] megmutatta a hierarchikus MOGA előnyeit a korábbi kutatásaikhoz képest, amely felhasználásával szignifikáns javulást tapasztaltak a klaszterezési eljárásuk során. Bujlow és tsai. [17] megmutatta, hogy a C5.0 mesterséges intelligencia eljárást miként lehet alkalmazni a hálózati forgalom osztályozására. A kutatásuk taglalja a pontos forgalmi adatgyűjtés lehetőségeit, ismerteti az osztályozási folyamat során felhasznált áramlási jellemzőket és bemutatja a C5.0 algoritmus alkalamzási lehetőségeit, végül pedig értékeli és összehasonlítja a kapott eredményeket. Az eljárás pontossága elérte a 99,3-99,9%-os intervallumot hét különböző alkalamzás megkülönböztetése esetén.

Fontos következtetés, hogy a feljebb részletezett, két megközelítésbe sorolt kapcsolódó munkák közül az első csoport az optimális a torlódás detektáció előrejelzését tekintve, mivel azok az alkalmasabbak prevenciós lépések megtételére a szolgáltatásromlás elkerülése, vagy legalább mérsékelése érdekében.

A hálózati adatfolyamok jellemzőinek függvényében ugyancsak két csoportra bonthatóak a vizsgálati folyamatok. Az elsődleges és relevánsabb megközelítés szemszögéből a folyamok korai áramlási jellemzőinek figyelembevételével történik a statisztikák kiszámítása illetve a csomag jellemzők felhasználása [7], [8], [10], ezek a jellemzők pillanatnyi adott csomagra vonatkozik a vizsgált adatfolyamok teljes képe nélkül. Míg az utólagos feldolgozás esetén [12]–[17] kumulatív statisztikai jellemzők állnak elő, hiszen az adatfolyamok teljes csomagszámjára nézve tudják leképezni a statisztikai mutatókat. A két megoldás természetesen kombinálható és ennek létjogosultságát több munkában is vizsgálták [9], [11], e kettő megközelítés egyidejű alkalmazásának köszönhetően egy nagyobb pontosságú megoldáshoz jutottak a problémakör kutatói.

A dolgozatban bemutatom a korai áramlási jellemzőket, ezeket felhasználva a kommunikációs tér felbontását horizontális és vertikális megközelítésből. Meghatározom továbbá a szolgáltatásromláshoz kapcsolódó metrikákat és algoritmikus lépéseket, amire használati esetet mutatok be. A horizontális és vertikális szolgáltatásromlás észlelése esetén az algoritimikus eljárást bevezetem, amely a Gamma eloszlással aktualizált maximum likelihood eljárásból származtatódik. Végül pedig gyakorlati validálását teszem meg a két különböző megközelítésnek, ahol egy vizsgálati adathalmazon való futás utáni statisztikai mutatók alapján vonok le következtetéseket. Ezen következtetésekből megállapítom az optimális n küszöbértéket, valamint a horizontális és vertikális szolgáltatásromlás észlelést elemzem ezen peremfeltétel mellett.

2. fejezet

Módszertan

Ahhoz, hogy a problémát egzaktul definiálni tudjuk meg kell vizsgálni az egyes szegmenseit az adatfolyamoknak. A következő szekcióban taglalom az egyes felbontási lépéseket, illetve lehetőségeket ahhoz, hogy a szolgáltatásromlás észlelés megvalósítható legyen a forgalom korai áramlási jellemzőinek felhasználásával.

Minden adatfolyam rendelkezik egy vertikális és horizontális felbontással, illetve egy időbeli kiterjedéssel. A vertikális felbontás nem más mint a kommunikáció irányultságának való taglalása a forrás azonosításának függvényében. Ezt részletesen a 2.3. fejezetben mutatom be. A horizontális felbontás képviseli a megfigyelt és nem-megfigyelt csomagok szerinti szeparációját az adott adatfolyamnak, amelyről a 2.2. fejezetben adok részletes bemutatást. Az időbeli kiterjedés az egyes adatcsomagokból tevődik össze, melyet a 2.4. fejezetben taglalok.

2.1. Korai áramlási jellemzők

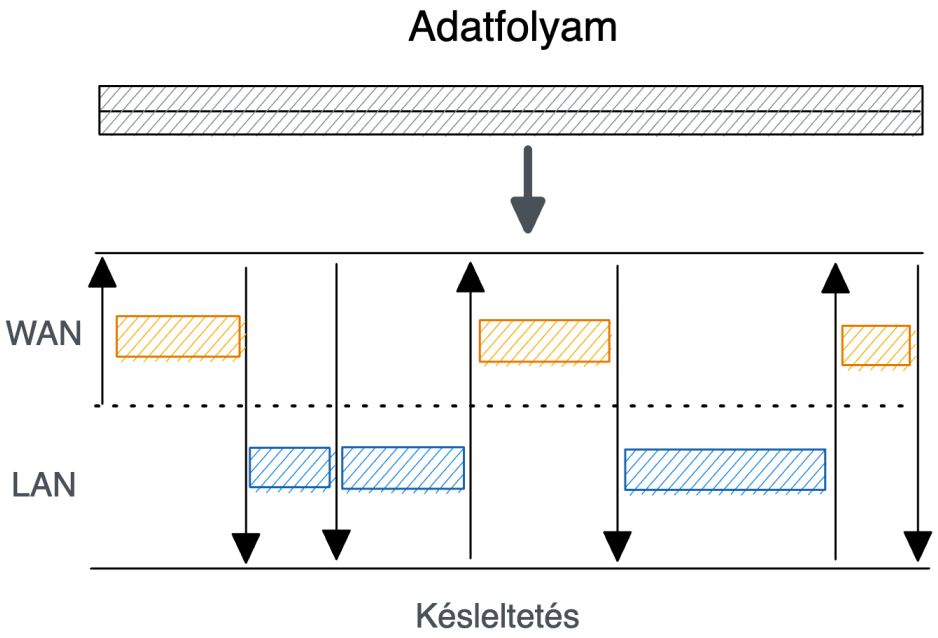
A korai áramlási jellemzők a megfigyelésre alapuló analitikai mutatók. Ezek az értékek a megszokott hálózati adatfolyam jellemezőktől nagyban eltérnek, hiszen ezeket az időbeli kiszolgálási lehetőségek befolyásolják. Egy általános adatfolyam jellemző esetén mint a forrás IP cím vagy protokol a környezet és időbeliség nem tud változtatni egy adott kérés esetén, hiszen ezek reprodubilis tulajdonsággal rendelkeznek a determinizmusuk miatt. Ezzel ellentétben az érkezési idők közét, vagy az adatcsomag méreteket nem tudjuk determinisztikusan reprodukálni ugyanazon hálózati körülmények mellett feltéve természetesen, hogy nem labori zárt körülmények között vizsgáljuk a hálózatot, hanem valós tényleges interneten történő aktív hálózati folyamok mellett végezzük a szimulációt.

Három korai áramlási jellemzőt fogok a munkám során felhasználni amelyek a következőek: *(i)* Az adott adatfolyam csomagjainak direkcióját az **splt_direction** listában gyűjti a hálózati adat analízisre szolgáló keretrendszer. Három különböző értéket azonosítunk az irányt tekintve. Abban az esetben ha 0-ás elem helyezkedik el a lista adott pozíciójában, akkor azt forrásból a célállomás felé irányuló csomagnak tekintjük. Amikor 1-es elem látható, akkor a célállomás irányából a forrás felé utazik a csomag. A -1 esetén pedig nem történt csomagküldés az adatfo-

lyam ezen állapotában. *(ii)* A csomagméret értékeket a **splt_ps** listában tárolja a keretrendszer, a csomagméretet pedig bájtban értendő. Ugyancsak elképzelhető itt is, hogy nem történt csomagküldés, ekkor a méretét sem tudjuk értelmezni a hiányzó csomagnak, ekkor a keretrendszer -1-el jelöli a csomaghiányt. *(iii)* A csomagok érkezési idejét végül a **splt_piat_ms** listában kerülnek letárolásra, amelynek a mértékegysége miliszekundumban értendő. Az adatfolyam első csomaga esetén ez érték mindig 0-ás értéket vesz fel, illetve a hiányzó csomag esetén a korábbiakhoz hasonlóan -1-el kerül jelölésre a csomag defilicit.

2.2. Kommunikációs irányultság felbontása

A kommunikációs irányultságnak két fajtáját különítjük el. Egyik a LAN oldali, míg a másik a WAN oldali direkció. Elkülönítésük relevanciáját a statisztikák helytállósága igényli, hiszen a WAN oldalon fellépő nem az ISP-től függő körülmények negatívan tudják befolyásolni az eredményeket. Ezt a fajta szétbontását a csomag irányultságának vertikális felbontásnak hívom továbbiakban a munkám során. A 2.1. ábrán vizuálisan látható, hogy egy adott véges adatfolyam miként tevődök össze WAN (amely sárga színnel van jelölve) és LAN (amit a kék szín ábrázol) késleltetésből. A dolgozat során a szolgáltatásromlás detektáció csakis az ISP belső infranetjére vonatkozik, ebből kifolyólag a továbbiakban a LAN oldali (kék színnel jelölt) késleltetések elemzésével és vizsgálatával fogok foglalkozni.



2.1. ábra. Egy adatfolyam vertikális felbontása.

A 2.1. fejezetben bemutatott jellemzőkben láthatóak alapján felhasználom a direkciót azonosító változót. A 1. eljárás bemutatja a csupán LAN késleltetésekből álló lista előállítását. Az algoritmus előkövetelménye, hogy a korábban előírt korlátokat teljesítse a felhasználandó jellemző.

1. Algorithm Kommunikációs irányultsági felbontás.

```
1:  procedure COMMUNICATION_DECTION_SPLIT(splt_direction, splt_ps)
2:      lan_delays = []
3:      previous_direction = -1
4:      for index, value in enumerate(splt_direction) do
5:          if index == 0 then
6:              previous_direction = direction
7:          else
8:              if direction == 0 and previous_direction == 1 then
9:                  lan_delays.append(splt_ps[index])
10:             end if
11:             previous_direction = direction
12:         end if
13:     return lan_delays
```

Az eljárás végig iterál a korai jellemző listáján és megvizsgálja, hogy történt-e változás a címkézésben a két állapot között. Abban az esetben, ha a korábbi címke 1-es jelzést kapott és az aktuálisan vizsgált címke 0-ás abból egyértelműen következik, hogy a vertikális lépésben egy LAN oldali késleltetés fog bekövetkezni.

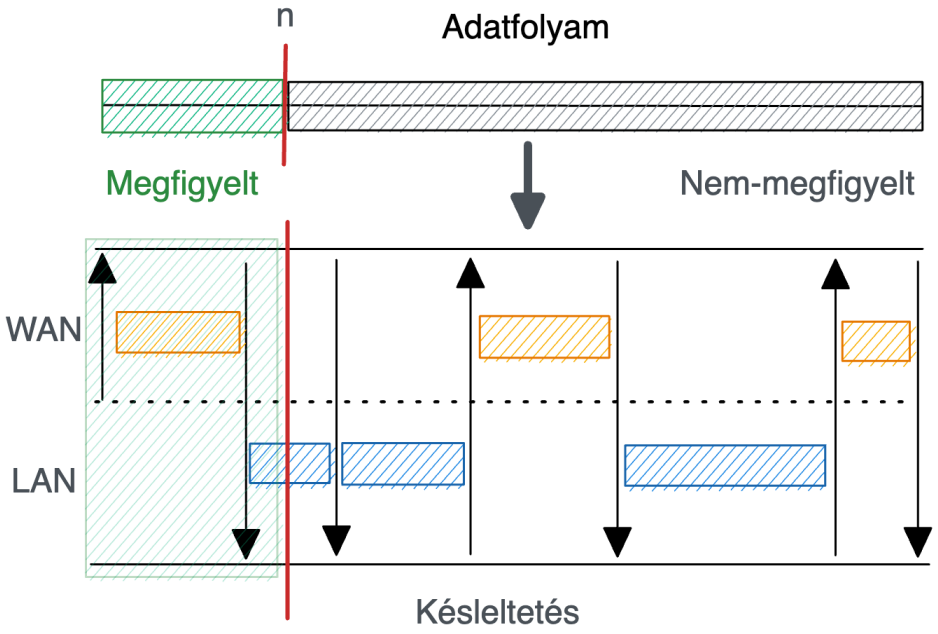
2.3. Kommunikációs tér felbontása

A kommunikációs tér felbontása vagyis a megfigyelt és nem-megfigyelt csomagok egy adott adatfolyam esetén való elszeparálása az egyes adatfolyamok véges csomagszámán értendő. Az n paraméter meghatározza, hogy az adott adatfolyam hány darab csomagját tekintjük megfigyeltnek. Több eset tud előállni ezen paraméter miatt, amelyek a következőek: *(i)* Elképzelhető, hogy az adott adatfolyam mérete (vagyis a csomagok száma) kisebb (vagy egyenlő azzal) mint a korlátényező, azaz $||f_i|| \leq n$, ahol f_i az i.-edik adatfolyamot jelenti. Ebben az esetben az összes jelenlévő csomagot megfigyeltnek tekintjük. *(ii)* Ugyancsak előállhat az eset amikor $||f_i|| > n$, ekkor a megfigyelt elemeket O és a nem-megfigyelt elemeket NO-val jelöljük. Az egyes csomagokat p_j-vel jelölöm, ahol j határozza meg az időrendi sorrendiségben az elhelyezkedését az adott csomagnak. Ekkor az f_i az alább két valós részhalmazra bontható $O = \{p_1, \ldots, p_j\} \subseteq f_i$ és $NO = \{p_{j+1}, \ldots, p_{||f_i||}\} \subseteq f_i$.

A 2.2. ábrán látható a *(ii)*-ben leírtak vizualizálva. Fontos megjegyezni, hogy az n paraméter nem kezeli külön a LAN és WAN oldali kommunikációs irányokat, amiből következik az optimális paraméter megtalálásának a nehézsége, hiszen nem tudjuk egzaktul garantálni az O halmazben szereplő LAN késleltetések darabszámát.

A *(ii)* esetben felléphet továbbá az eset amikor a megfigyelt és nem-megfigyelt határán egy váltás történik a 2.2. fejezetben bemutatottak szerint. Ilyenkor ez az adat elvész és ezt metrikát külön vizsgálom az egyes korlátényezők változásának függvényében.

A két halmaz adatainak előállításához iteratívan szükséges futtatni a 2. eljárást külön-külön a megfigyelt és nem megfigyelt részhalmazokra az alábbiak szerint. Már itt látható a ténye,



2.2. ábra. Egy adatfolyam horizontális felbontása.

hogy az *i* paraméter függvényében elvágható egy-egy LAN direkció váltás amivel csökken a LAN oldali késleltetések számának szumma összege a megfigyelt részre vetítve.

Az ideális *n* paraméter megtalálása egy komplex feladat, hiszen nem tudunk univerzális képet alkotni minden egyes adatfolyam típusra applikáció kategória függetlenül, hogy mennyi és milyen direkciójú csomag fogja alkotni azt.

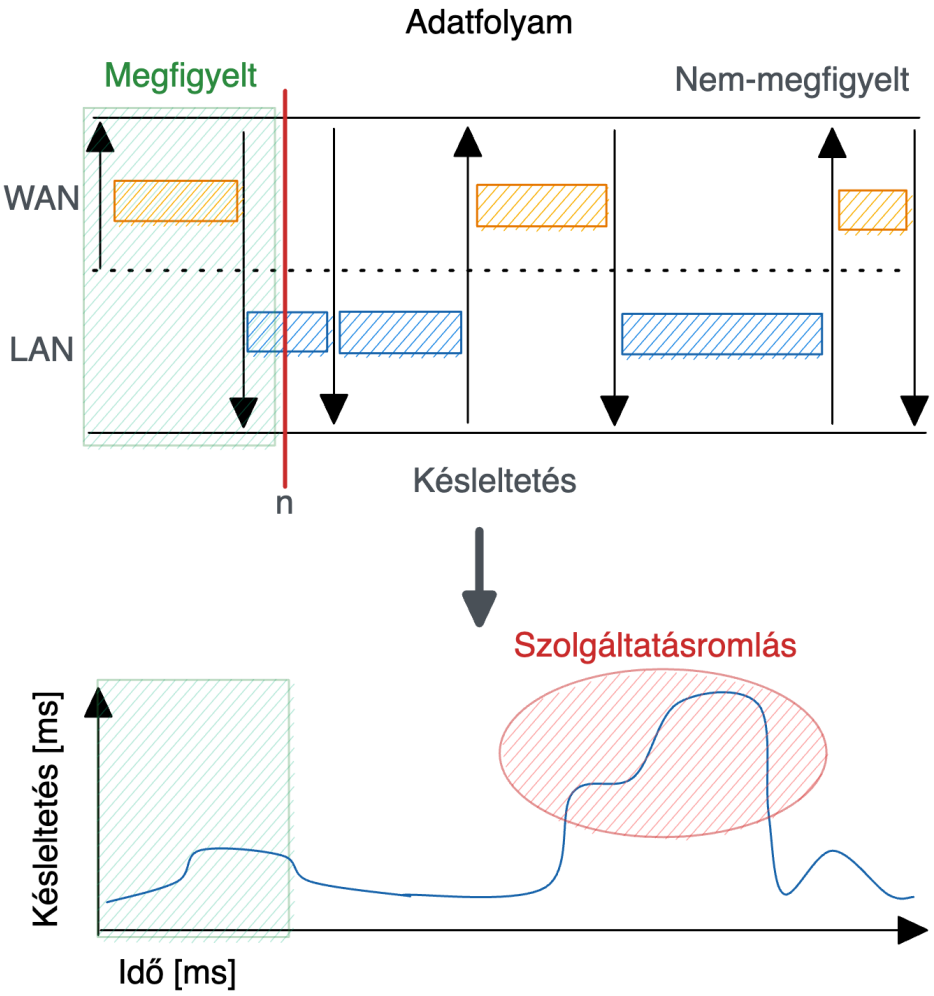
2. Algorithm Kommunikációs tér felbontás.

```
observed_lan_delays = []
non_observed_lan_delays = []
for i in range(1, ||splt_ps||) do
    o_sdg = communication_dection_split(splt_direction[:i], splt_ps[:i])
    observed_lan_delays.append(o_sdg)
    no_sdg = communication_dection_split(splt_direction[i:], splt_ps[i:])
    non_observed_lan_delays.append(no_sdg)
```

2.4. Szolgáltatásromlás detektálása

A szolgáltatásromlás észlelésének definiálása nehézkes és az adott szolgáltatás típusok függvényében eltérő, hiszen egy késleltetés érzékeny alkalmazás esetén minimális késleltetésbeli különbség esetén is szolgáltatásromlásról beszélünk mint egy hanghívás, vagy videó streaming. Ezzel ellentétben a késleltetés növekedése egy letöltés vagy e-mail küldés során nem észrevehető, hiszen ezek nem érzékenyek a késleltetés szemszögéből. Ezáltal a szolgáltatásromlás egy applikáció kategóriától függő küszöbértékkel rendelkezik, hogy mekkora torzulást enged a szállító médium paraméterein.

A 2.3. ábrán a szolgáltatásromlás a piros karikában látható kilengése az *y* tengely mentén, ahol a késleltetés miliszekundumban van ábrázolva. A dolgozat során egységesen szolgáltatásromlásnak tekintek minden olyan esetet, amikor a késleltetésben a megfigyelt állapothoz (zöld téglatestben elhelyezkedő görbe alatti terület) képest a nem-megfigyelt állapotban jelentős eltérés látható.



2.3. ábra. Egy adatfolyam esetén fellépő szolgáltatásromlás.

Az alapfeltevés a szolgáltatásromlás meghatározásához a megfigyelt és nem-megfigyelt halmazok késleltetésének kommunált átlagán alapul, amely az alábbiak szerint tevődik össze,

$$\frac{1}{||D^O||} * \sum_{n=1}^{||D^O||} D_n^O < \frac{1}{||D^{NO}||} * \sum_{n=1}^{||D^{NO}||} D_n^{NO} * (1.0 + \varepsilon) \tag{2.1}$$

ahol D^O a megfigyelt oldali késleltetésket tartalmazza, míg D^{NO} a nem-megfigyelt oldali késleleltetéseket és a ε futóparaméter pedig az a küszöbérték ami meghatározza mekkora eltérést enged meg a két állapot átlaga között.

Felmerül természetesen a kérdés, hogy mi igaz abban az esetben, ha az egyenlőtlenség baloldalán álló kifejezés jelentősen nagyobb mint a jobboldalán látható. Ilyenkor a szolgáltatás

javulást fogja a végfelhasználó érzékelni és mivel ez a megfigyelt állapotába került a halmazoknak így ezzel külön nem foglalkozom a dolgozat során, mert ezt már a jelenleg is meglévő megoldás le tudja kezelni abban az esetben, ha a többi adatfolyam megfigyelt állapotához képest jelentős eltérés van. Ennek a meghatározásának a terminológiája teljesen azonos a 2.1. egyenlettel.

Ahhoz, hogy a szolgátatásromlás pontos meghatározása megtudjon történni elengedhetetlen további összefüggések keresése maga az adatfolyamok között a bemutatott eljárásokat alkalmazva. A jelentős eltérés pontos definiálásához további különböző eljárásokat elemeztem és alkalmaztam, amelyek részletes bemutatását a 2.5. fejezet tárgyal, hiszen önmagában ezen bemutatott faktum kevés egy predikciós eljárás megalkotásához.

2.5. Használati eset

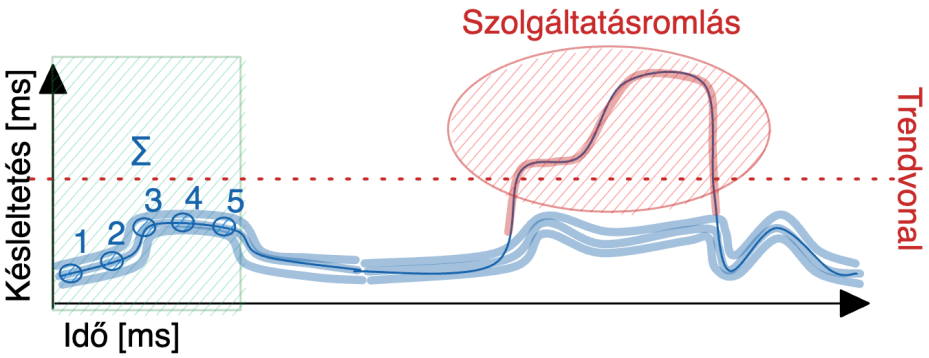
A következőben a 2. fejezetben bemutatott módszertan gyakorlati alkalmazhatóságára teszek két javaslatot. Az alkalmazások bemutatják miként lehet szolgáltatásromlás észlelését végezni a hálózati adatforgalom korai áramlási jellemzőinek figyelembevételével.

Két különböző megközelítést fog a fejezet taglalni. Először a 2.5.1. fejezetben a dolgozat bemutatja miként lehetséges a szolgáltatásromlás detektálása horizontálisan, vagyis a megfigyelt és nem-megfigyelt halmazok közötti korrelációt. Ennek hála alkalmasan megtudjuk választani a megfigyelt és nem-megfigyelt halmazokat szétválasztó paraméter értékét, hogy a lehető leginformatívabb statisztikákat tudjanak az ISP-k előállítani. Ezek után a 2.5.2. fejezetben megmutatom miként lehet vertikálisan felhasználni a megfigyelt állapotban a korai áramlási jellemzőket felhasználva egy szolgáltatásromlás észlelést végezni más adatfolyamokra kivetítve úgy, hogy azok időben eltolódnak a vizsgált adatfolyamhoz képest. Ennek köszönhetően további mélyebb elemzést biztosítok az elválasztó paraméter meghatározása terén, illetve az egyes adatfolyamok együttes kihatását vizsgálom a végfelhasználóra. Fontos látni, hogy milyen hatással van egy megfigyelt állapotban lévő adatfolyamban fellépő szolgáltatásromlás más adatfolyamokra főként, ha azok nem-megfigyelt állapotukban helyezkednek el, hiszen ezáltal tudunk referálni a megfigyelt szolgáltatásromlás alapján más adatfolyamok viselkedésére. Az ISP-k ezen tudás felhasználásával tovább tudják pontosítani az előrejelzési mechanizmusukat és így egy robosztusabb szolgáltatásmenedzsment (QoS) kialakítása is elképzelhetővé válhat.

2.5.1. Horizontális szolgáltatásromlás észlelése

A horizontális szolgáltatásromlás detektációjához meg kell vizsgálni a megfigyelt állapotban, illetve a nem-megfigyelt állapotban mért késleltetés értékeket. Ebből is látható, hogy mekkora kihatással van a hátérték pontos meghatározása, hiszen az együttes kialakult átlagot nagymértékben befolyásolni tudja a szolgáltatásromlás detektációt. A megfigyelt állapotban mért késleltetések és obszervált mintázatok az adott adatfolyam trendvonala. A 2.4. ábrán látható a legegyszerűbb példa, amikor a trendvonal megállapítása tisztán a megfigyelt állapot számtani

közepe alapján határozom meg. Ennél már egy fokkal szofisztikáltabb megoldás, ha az ε sugarú hibát is figyelembe vesszük, de még mindig nagyon kezdetleges megközelítés.



2.4. ábra. Trendvonal meghatározása egy adatfolyam esetén.

Az adatfolyamok késleltetése a Gamma eloszlást követik [18], [19]. A becslési eljáráshoz a dolgozat a maximum likelihood módszerét alkalmazza, amely szélsőérték feladatot a Gamma eloszlásra kell felírni. A Gamma eloszlásról ismert, hogy sűrűségfüggvénye

$$f(x) = \frac{\lambda^p * x^{p-1} * e^{-\lambda * x}}{\Gamma(p)}, \tag{2.2}$$

ahol $\Gamma(p)$ a p pozitív paraméterű gamma-függvény. A likelihood-függvény pedig az alábbiak szerint irható fel, ahol a rögzített q paraméter mellett tetszőleges $x_1, \ldots, x_n$ értékkel határozható meg a sűrűségfüggvény az alábbiak szerint:

$$L(q) = \prod_{i=1}^{n} f_{x_i}(x_i; q). \tag{2.3}$$

Ahhoz, hogy megkapjuk a likelihood-függvényt a Gamma eloszlást feltételezve, amely leírja megfelelően az egyes adatfolyamok késleltetésének viselkedését, ahhoz a 2.3. függvénybe szükséges a 2.2. egyenletet behelyettesíteni. A probléma a következőképpen írható ekkor fel:

$$L(p, \lambda | \mathbf{x}) = \prod_{i=1}^{n} \frac{\lambda^p * x_i^{p-1} * e^{-\lambda * x_i}}{\Gamma(p)}. \tag{2.4}$$

Feltéve, hogy $p > 0$ és $\alpha > 0$ teljesül. Hasonlóan természetesen a loglikelihood függvény is előállítható, ha vesszük a 2.4. egyenlet természetes logaritmusát.

A maximum likelihood függvény Gamma eloszlásra aktualizált egyenletét felhasználva szükséges egy becslést előállítani a megfigyelt állapotban lévő adatpontokra, látható, hogy az n küszöb paraméter meghatározása mennyire fontos, hiszen a függvénygörbe ívének Gamma eloszlás jellegű lecsengését nagyban befolyásolja a mintavételezett késleltetések. A kiszámításnak menete az alábbiak szerint történik:

Ezen procedúra szerint előállt vektorok alapján illesztésre kerül a mintavételezett csomagok korai áramlási jellemzőiben fellépő késleltetés és az eltérés esetén címkézésre kerülnek az egyes

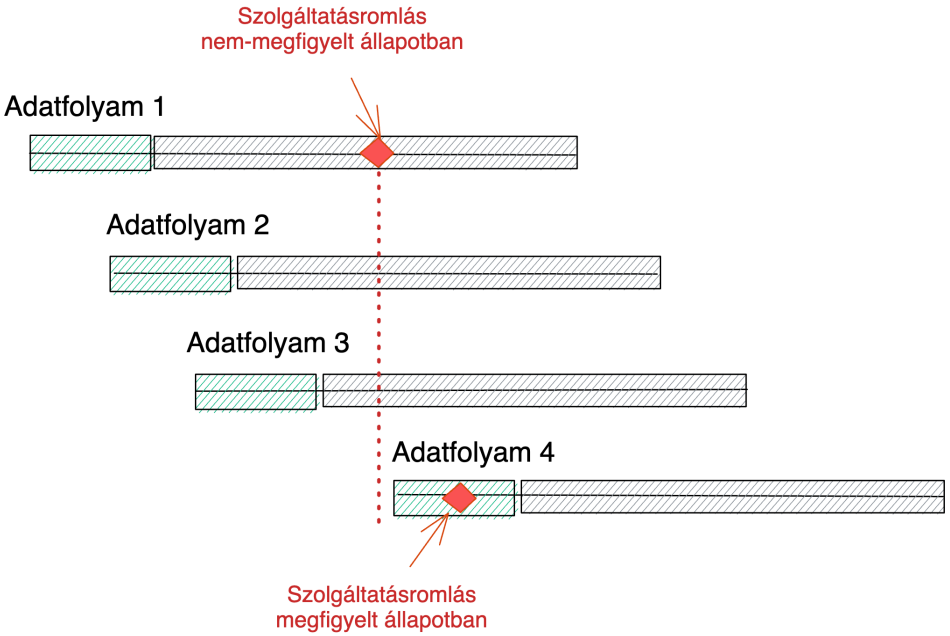
3. Algorithm Maximum likelihood számítás.

```
1: procedure MAXIMUM_LIKELIHOOD_GAMMADISTRIBUTION(X)
2:     mean = ∑_{i=1}^{||X||} X_i × ||X||^{-1}
3:     variance = ∑_{i=1}^{||X||} (X_i − μ)^2 × N^{-1}
4:     alpha = mean^2 × variance^{-1}
5:     beta = alpha × mean^{-1}
6:     return X.map(lambda x: gamma.pdf(x, alpha, loc=0.0, scale=1.0/beta)).prod()
```

adatfolyamok a fellépő szolgáltatásromlás helyének figyelembevételével. A szolgáltatásromlás helye alatt a csomagszám és az időintervallum együttese értendő. A vektorok előállítása az n paraméter által meghatározott csomagok LAN oldali késleltetése által áll elő. Ahhoz, hogy később a hatástanulmány során elemezhető legyen a paraméter konfiguráció szemszögéből iteratív módon növekvő n értékre előállításra kerül a szolgáltatásromlás megállapítása a helyazonosító listák fenntartásával.

2.5.2. Vertikális szolgáltatásromlás észlelése

A vertikális szolgáltatásromlás észlelés során azon esetek kerülnek taglalásra, amikor egy adott adatfolyam nem-megfigyelt állapotában fellépő szolgáltatásromlás milyen implikációkkal rendelkezik más adatfolyamok megfigyelt állapotára. Ezen két vagy több adatfolyam közötti reláció előfeltétele viszont, hogy időrendiség szemszögéből rendezett adathalmaznak tekintsük az adatfolyamokat. Ez a típusú leképzés két adatfolyam között a 2.5. ábrán látható.



2.5. ábra. Két adatfolyam közötti koherencia.

A megoldáshoz meg kell találni azt az optimális k értéket amely meghatározza az időrendi sorrendben beérkező adatfolyamok vizsgálati számát. Ez szükséges a kontrollációhoz, hogy mi-

lyen gyakorisággal kerüljön a memóriából kiürítésre a vizsgált adatfolyam. Kardinális kérdés ezen szám meghatározása, hiszen alulméretezés esetén nem kerülnek leképzésre az egyes implikációk ezáltal nem optimális a működés, míg túlméretezés esetén addicionális erőforrásigény jelenik meg, amely egy otthoni hálózati végfelhasználói eszköz esetén az ISP-ket túl nagy anyagi járadékkal terhelné. A memóriaigény további befolyásoló tényezője természetesen az egyes adatfolyamok esetén megtartandó n darab csomag korai áramlási jellemzőinek tárhelyigénye. E kettő érték együttese adja meg az eljárás teljes helyigényét. Ennek a méretezése és a fizikai hardverre történő specifikuma nem képezi ezen dolgozat részét, továbbá azzal a feltételezéssel élek, hogy ezen adatmennyiség az átlag hálózati útvonalválasztó eszköz memóriakorlátaiba belefér.

A 4. eljárásban bemutatott algoritmus a 2.5.1. fejezetben megmutatottak alapján megvizsgálja, hogy az egyes adatfolyamban fellép-e a nem-megfigyelt állapotban szolgáltatásromlás, majd ezek után a következő k adatfolyam (időrendileg rendezett halmazon értelmezve) esetén az eljárás megvizsgálja a megfigyelt állapotban fellépő szolgáltatásromlást. A cél ezzel, hogy két adatfolyam esetén milyen fellépő indikációja van egy megjelenő szolgáltatásromlásnak. Az eljárás által rámutathatunk arra a tényre miszerint az egyes adatfolyamok a hálózati eszközben nem független entitások és együttes vizsgálatuk alapján állapítható meg a teljeskörű kép a hálózati eszköz teljesítményéről.

4. Algorithm Vertikális szolgáltatásromlás detektációs eljárás.

```
1:  for index, row in df.iterrows() do
2:      for sdno_index, sdno_identifier in enumerate(sd_in_no) do
3:          shifted_time = row['bidirectional_first_seen_ms'] + elapsed_time[sdno_index]
4:          for index_other, row_other in islice(df.iterrows(), index, None) do
5:              if shifted_time ≤ row_other['bidirectional_first_seen_ms'] then
6:                  for sdo_index, sdo_identifier in enumerate(sd_in_o do
7:                      if sdo_identifier == 1 then
8:                          service degradation identified
9:                      end if
10:             end if
```

Természetesen az eljárás előkövetelménye, hogy időrendileg rendezett legyen az adathalmazunk, így megkönnyítve az iterálást az egyes vizsgált adatfolyamokon. Mint látható a bemutatott 4. eljárás nem egy online algoritmus, viszont az egyszerűbb tárgyalhatóság miatt a nem effektívebb eljárás bemutatása célratörőbb. Komplexitását tekintve az eljárás M darabszámú adatfolyam esetén, ahol n különböző megfigyelt/nem-megfigyelt állapotvágást teszünk, így legrosszabb esetben az adatfolyamokból előáll egy $O(M^2)$, amelyhez kapcsolódik az n paraméter függvényében addicionálisan $O(n^2)$ komplexitás. A teljes eljárásra ez alapján a kettő kommunált komplexitását tudjuk mondani ami nem más mint $O(M^2+n^2)$

Az eljárás ideálisan rámutat arra a tényre, hogy szükséges a korai áramlási jellemzők alapján a szolgáltatásromlás detektálására és a QoS adaptív finomítására, hiszen az egyes adatfolyamok koherenciába állnak egymással, így egy kedvezőtlen helyzetben, ha ezt a tényt figyelmenkívül

hagyjuk akkor a szolgáltatásromlások egymásra vetített negatív hatásai miatt az ügyfélelégedettség jelentősen romlik. Mindezek egy megfelelő a dolgozat keretén bemutatott eljárással elkerülhetőek lennének ideális paraméter konfiguráció mellett, mellyel az ügyfélelégedettség növekedni tud.

3. fejezet

Gyakorlati validálás

A következő fejezetben bemutatásra kerül az egyes eljárások által prognosztizált szolgáltatásromlások statisztikai előfordulása. Az ideális n paraméter megtalálásának érdekében történő átfogó tanulmányozást, amely optimalizálási művelet a LAN odali késleltetések számtani közepének függvényében, ahogy az a 2.4. fejezetben bemutatásra került. A vizsgálatok során felhasznált adathalmazzal kapcsolatos információk, ezentúl az egyes irányultságú szolgáltatásromlások elemzései is megvizsgálására kerülnek.

3.1. Vizsgálati adathalmaz

Az eljárások vizsgálatához egy átlagos méretű európai egyetem hálózatának adatai kerültek felhasználásra. Az átlagos forgalom az uplink-en napi szinten átlagosan 1,6 Gbps körüli értéket mutatott. A mérés teljes időtartalma 299 másodperc, amely magába foglal 62 millió adatfolyam bejegyzést az összes áramlási jellemzőjével, amelyeket a tanulmányhoz elengedhetetlen fontossággal bírnak. Természetesen a mérés során futási időben anomalizáció történt azon mezőértékekre nézve, amely alapján a kliensek beazonosíthatóak lennének, így nem sértve a személyi jogaikat a felhasználóknak. A mérés során a hálózati link bitrátája 1320 Mbps volt, míg a adatméretileg 165 Mbps. Az átlagos csomagméret 795,35 bájt és az átlagos csomagtovábbítási ráta 207.000 csomag/másodperc volt ezen feltételek mellett. Mindezek mellett fontos megjegyezni, hogy a COVID-19 járvány miatt a hálózati link nem érte el a tipikus forgalmi sebességet. Az adathalmaz főbb jellemzőit a 3.1. táblázat összesíti.

A dolgozat által taglalt egyes eloszlásfüggvény görbék esetén az x tengely mentén látható az aktuálisan vizsgált metrika, míg az y tengelyen látható a sűrűségfüggényen felvett valószínűségi változó a $[0;1]$ zárt intervallumon.

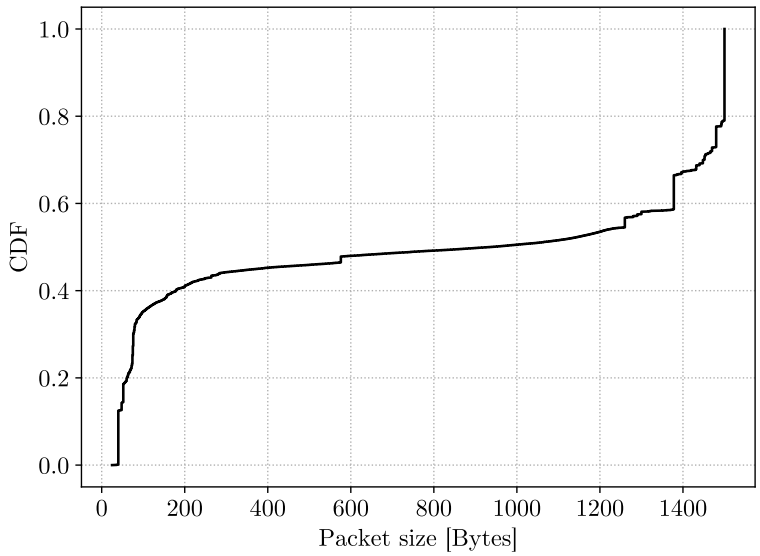
3.1.1. Csomagszintű kommunikációs jellemzők

Az eloszlásfüggvényét az egyes csomagoknak a 3.1. ábrán látható. Az általános célú MTU beállítások alatti méretet képviseli az összes csomag termsézetesen, viszont a csomagok 60% az 1200 bájt alatti méretű ami az UDP csomagok és a TCP fregmentációk következményében

3.1. táblázat. A vizsgálati adathalmaz csomagszintű jellemzői.

Csomagok száma	62 034 463
Adatméret	47 GB
Időtartam	298.94 másodperc
Első adatfolyam időbélyege	2021-05-13 10:24:14
Utolsó adatfolyam időbélyege	2021-05-13 10:29:13
Adatbájt átviteli sebesség	165 Mbps
Adatátivteli bitsebesség	1 320 Mbps
Átlagos csomagméret	795.35 Bytes
Átlagos csomagátviteli sebesség	207 kpackets/s

állnak elő. A vizsgálat során a csomagméretbe beszámításra kerültek az IP fejlécek is. A görbe karakterisztikája és a statisztikai mutatók alapján kijelenthető, hogy az adathalmaz esetén a legtöbb csomag IP-datagrammokat tartalmaznak.

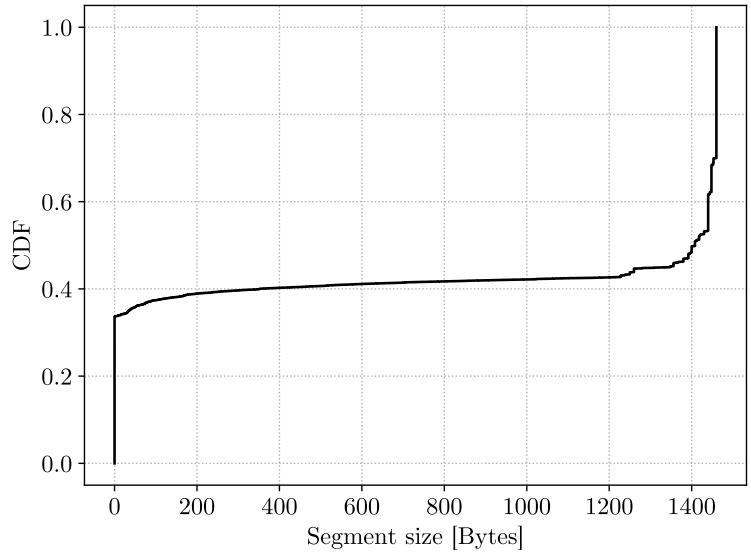


3.1. ábra. Adatcsomag méretek eloszlásfüggvénye.

A szegmensek méretének eloszlásfüggvénye a 3.2. ábra mutatja be. A csomagok 50% nem közelíti meg az MTU méretét ezáltal a szegmentálás szükséglete nem lép ezen esetekben fel. Míg a másik esetben a szegmentálás igénye felmerül, hiszen a tipikus 1500 bájtos MTU konfigurációhoz konvertálnak az adatkapcsolati rétegben utazó hasznos adat.

3.1.2. Folyamszintű kommunikációs jellemzők

A forgalmi nyomvonalakban szereplő csomagokat kétirányú áramlásokba rendeztük az NFStream eszköz segítségével, az alapértelmezett konfigurációs paraméterekkel, kivéve az üresjárati időt. Az áramlások üresjárati idejének lejárta azért szükséges, hogy két különböző áramlást ne lehessen azonosnak azonosítani. Tekintsünk két hosztot, ahol az egyik sok új TCP-kapcsolatot kezdeményez a másikhoz. Minden új TCP-kapcsolat új forrásportot kap az efemer portok egy



3.2. ábra. Szegmens méretek eloszlásfüggvénye.

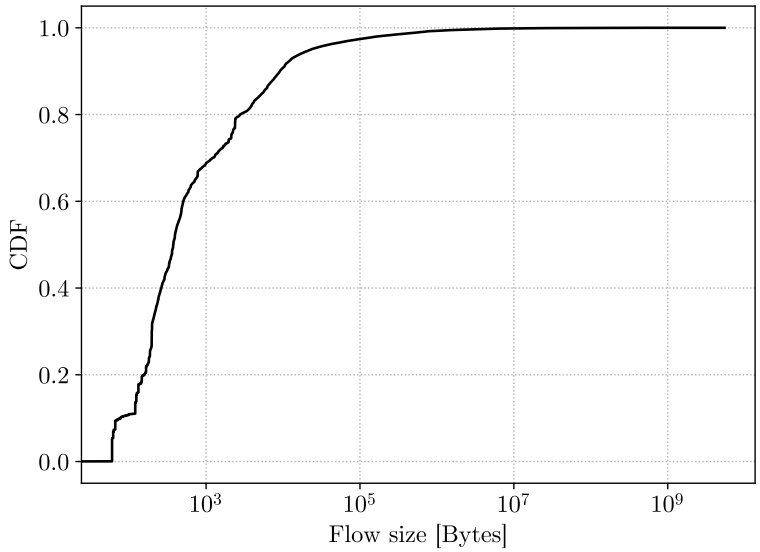
rögzített tartományából, amelyet általában eggyel növelnek az előző kiosztáshoz képest. Idővel a TCP-portok számai a tartományt átforgatják, és a legkisebb számmal kezdik újra. Ennek következtében ez problémát okoz, mivel egy új áramlásnak ugyanaz az 5-tuple-je lesz, mint egy korábbi áramlásnak. Ennek a problémának a megoldására az üresjárati idő használata az általános módszer. Az adatfolyamok csomag érkezési időközének szigorú elemzése után az üresjárati időt 120 másodpercre állítottam be. A 3.2. táblázat mutatja az egyes adathalmazokban kapott áramlások számát, valamint a TCP, UDP és egyéb protokollok közötti megoszlásukat.

3.2. táblázat. Leggyakrabbi protokollok listája.

Protokoll	**Folyam [∑]**
TCP	200 789
UDP	179 532
ICMP	48 078
IPv6-ICMP	114
HOPOPT	48
Egyéb	68
Összesen	428 629

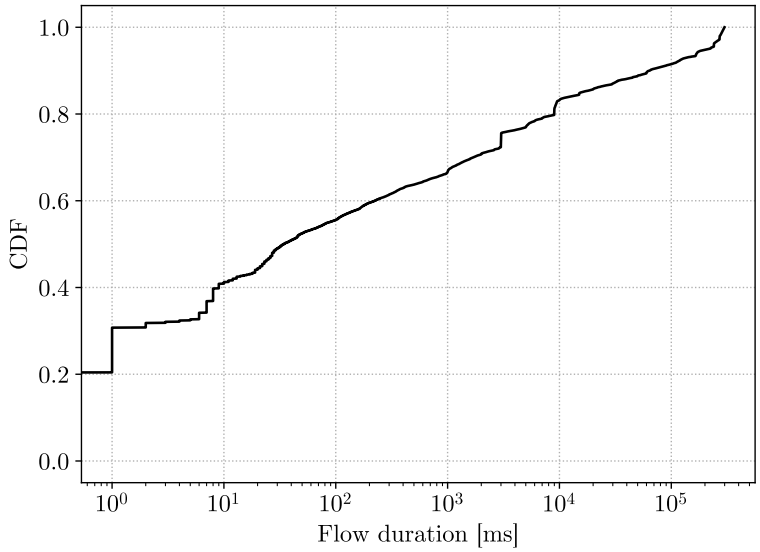
A 62 millió csomag esetén azonosításra került a TCP kapcsolat mint a legtöbb forgalmat generáló eljárás, ami összesen 200 789 adatfolyamot tett ki. A második legnagyobb kategória az UDP, amely 179 532 folyam esetén volt a támogatott eljárási protokoll. Az ICMP-n (amely a hálózati diagnosztikákhoz használt protokoll) kívül a további protokollok által generált forgalom elenyésző. A fő hálózati forgalom a TCP és UDP kapcsolatok keretén belül zajlik le. Fontos látni, hogy a TCP kapcsolat esetén egy háromutas kézfogással épül fel a kapcsolat amit az n paraméter vizsgálata esetén figyelembe kell venni, hiszen a tényleges szolgáltatásromlás detektálás nem tud megtörténni a kapcsolatfelépítési folyamat korai áramlási jellemzői alapján,

mivel ekkor még hasznos adat a fejlécben nem közlekedik. A hasznos adattal rendelkező folyam esetén kívül természetesen a kézfogásnál is feltud lépni szolgáltatásromlás a WAN oldal esetén, viszont a LAN oldalon ennek mérete elenyésző, így ezt külön ekkor nem érdemes vizsgálni.



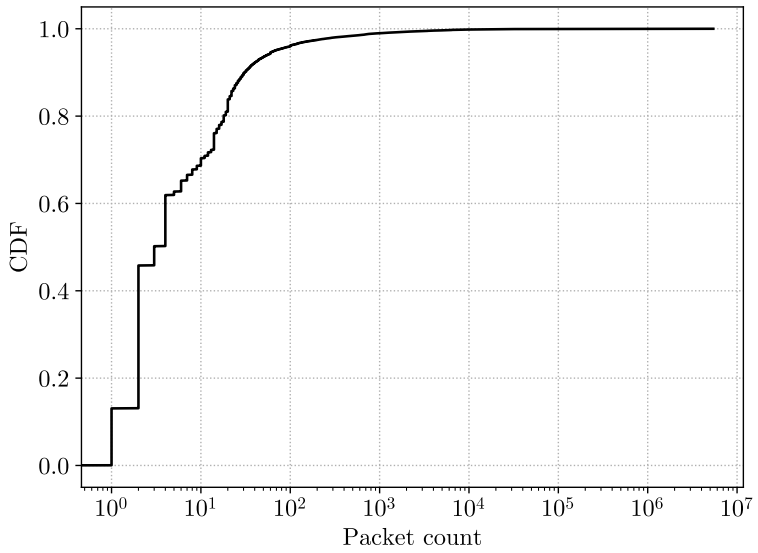
3.3. ábra. Adatfolyam méretek eloszlásfüggvénye.

Az adathalmazhoz kapcsolodó adatfolyam méretek eloszlásfüggvénye a 3.3. ábrán látható. Megállapítható, hogy az adatfolyamok 75%-a az általános méretű adatfolyam méretet képviseli és csak nagyob kevés esetben jelennek meg HH adatfolyamok. Az egyes adatfolyamok 80%-a a tíz kB-nál kisebb méretűek, míg a 20 kB és felette levő értékek a feső 3%-ban helyezkednek el. A mutatók alapján továbbá megállapítható, hogy azon áramlások, amelyek 100 vagy annál kevesebb csomagból tevődnek össze, generálják a teljes forgalom 98%-át.



3.4. ábra. Adatfolyam idővolumen eloszlásfüggvénye.

A 3.4. ábra megmutatja az adatfolyamok továbbításához tartozó időszükségletet. A csomagok felső 20%-ban 10 vagy annál több másodpercre volt szükség, hogy az adott adatfolyam összes csomaga továbbításra kerüljön. Míg túlnyomó többségben az esetek 58%-ban ez az idő nem haladta meg a 0,1 másodpercet.



3.5. ábra. Csomagszám eloszlásfüggvénye.

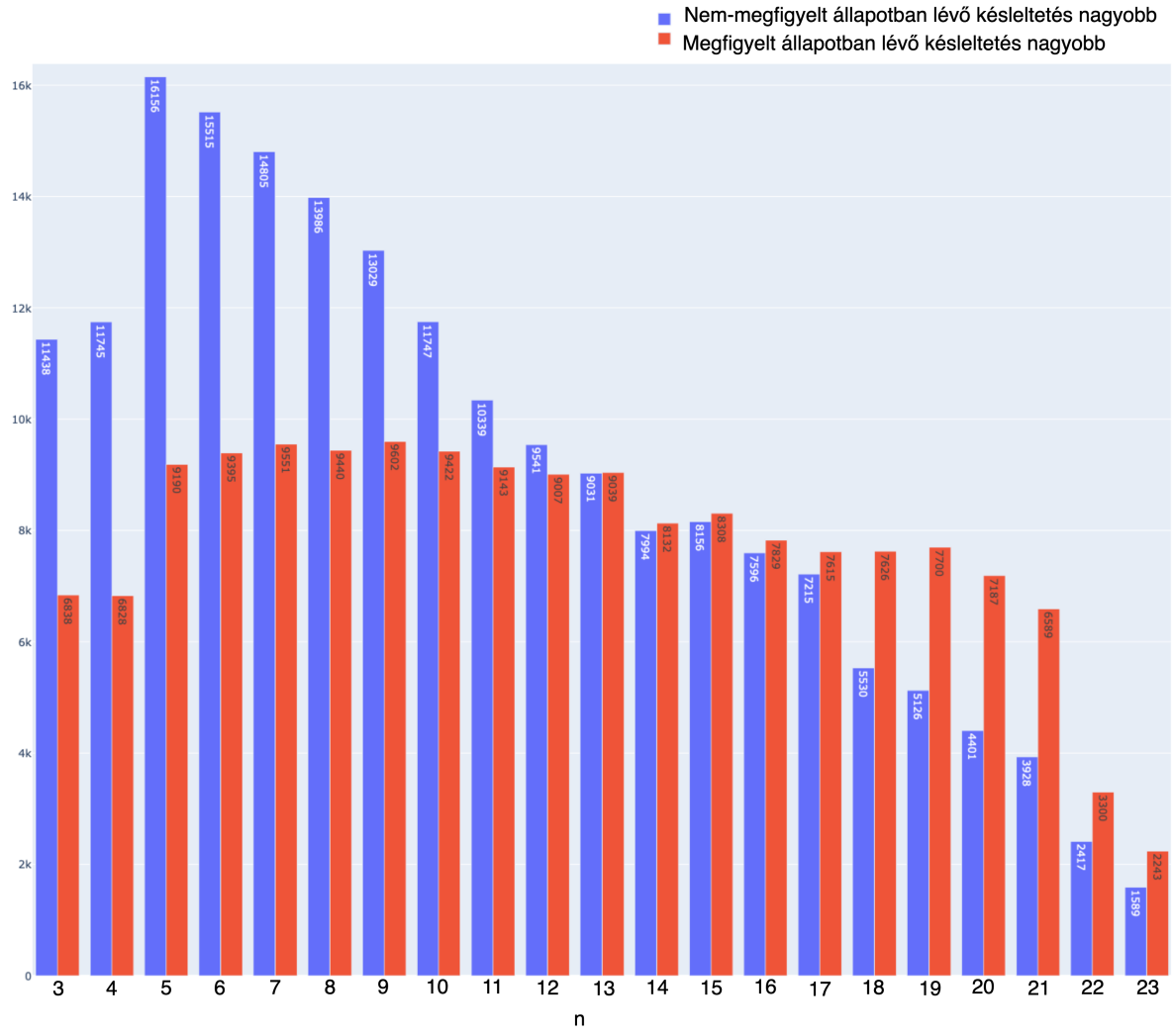
A 3.5. ábrán a csomagszám eloszlásfüggvénye alapján megállapítható, hogy az adatfolyamok felső 2%-a tevődik össze több mint 1000 csomagból, míg a maradék 98%-ban az adatfolyamok tipikusan 100-nál kevesebb csomagból állnak össze. Látható, hogy az adatfolyamokhoz tartozó egyes nézetek szoros kapcsolatban állnak egymással, hiszen a csomagok mérete az adatfolyamokban, illetve az időintervallumok és a szegmentálási információk függvényében teljeskörű képet lehet alkotni a vizsgálati adathalmazról.

Ezen adathalmazban megjelenő forgalmak megfelelően leírják egy nagyhálózati végfelhasználói csomópontot, ahol egyidejűleg megjelennek az egyes forgalmi adatosztályok mint a multimédiás-alkalmazások vagy a streaming szolgáltatások. A további elemezések keretén belül ezen adathalmaz kerül felhasználsára a dolgozat során.

3.2. Küszöbérték meghatározása

Az *n* küszöbérték megfelelő meghatározásához fontos megvizsgálni az egyes adatfolyamokra nézve a különböző *n* érték mellett mely állapotában nagyobb az átlagos késleltetés vagyis, hogy a felmerülő érték a megfigyelt vagy a nem-megfigyelt állapotban jelentősebb. A vizsgált adathalmazra előállt értékek kommunált képét kell vizsgálni a lehető legdeszkriptív eredményhez. A 3.6. ábrán látható az egyes paraméter érték esetén mekkora a kommunáltan a két halmaz méretének relációja.

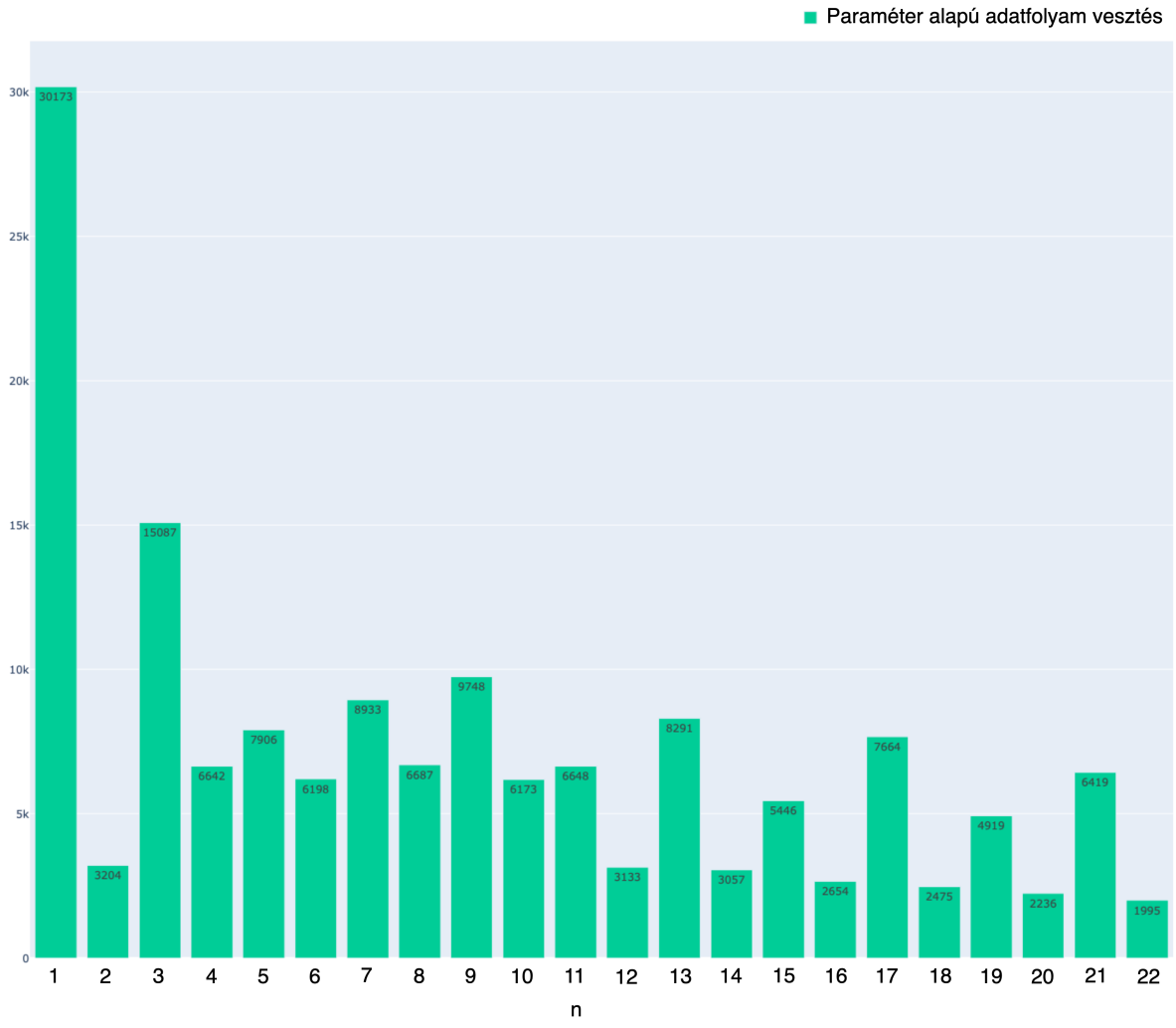
Kék színnel vannak jelölve azok az esetek, amikor a nem-megfigyelt állapotban lévő késlel-



3.6. ábra. Késleltetések relációjának kommunált értéke.

tetések értéke nagyobb volt mint a megfigyelt állapotban, és ennek inverzét pedig a piros szín szimbolizálja. Az ideális eset amikor a nem-megfigyelt állapotban lévő késleltetések száma (kék oszlop) maximalizálva van, míg a megfigyelt állapotban lévő késleltetések száma minimalizálva. Az ábra alapján jól látható, hogy ez a 5-11-es *n* paraméter konfiguráció mellett érhető el, hiszen ennél kisebb esetekben a paraméter miatt felmerülő vágási adatvesztés jelentkezik. A 13-as méret felett pedig, ahol az inflexió megtörténik és a megfigyelt oldal kerül maximalizálásra nem ideális a szolgáltatásromlás előrejelzésére. Fontos látni, hogy minden adatfolyam esetén az első 25 csomag került csakis vizsgálatra, hiszen az ennél több csomagból álló folyamok szorványosan fordulnak csak elő és a legnagyobb volument ezáltal nem ezek képezik. A továbbiakban az 5-11-es paraméter konfigurációk mentén kerül vizsgálatra a szolgáltatás romlás előrejelzés.

Korábban említésre került, hogy a paraméter által elválasztásra tud kerülni egy-egy megfigyelt állapotbeli átlépés, amely problémát tud statisztikai szemszögből okozni, amely a 3.6. ábrán jól megmutatkoznak a 3 és 4-es *n* paraméter konfiguráció mellett. Számszerűsítve ezek az esetek a 3.7. ábrán láthatóak zöld színnel jelölve.

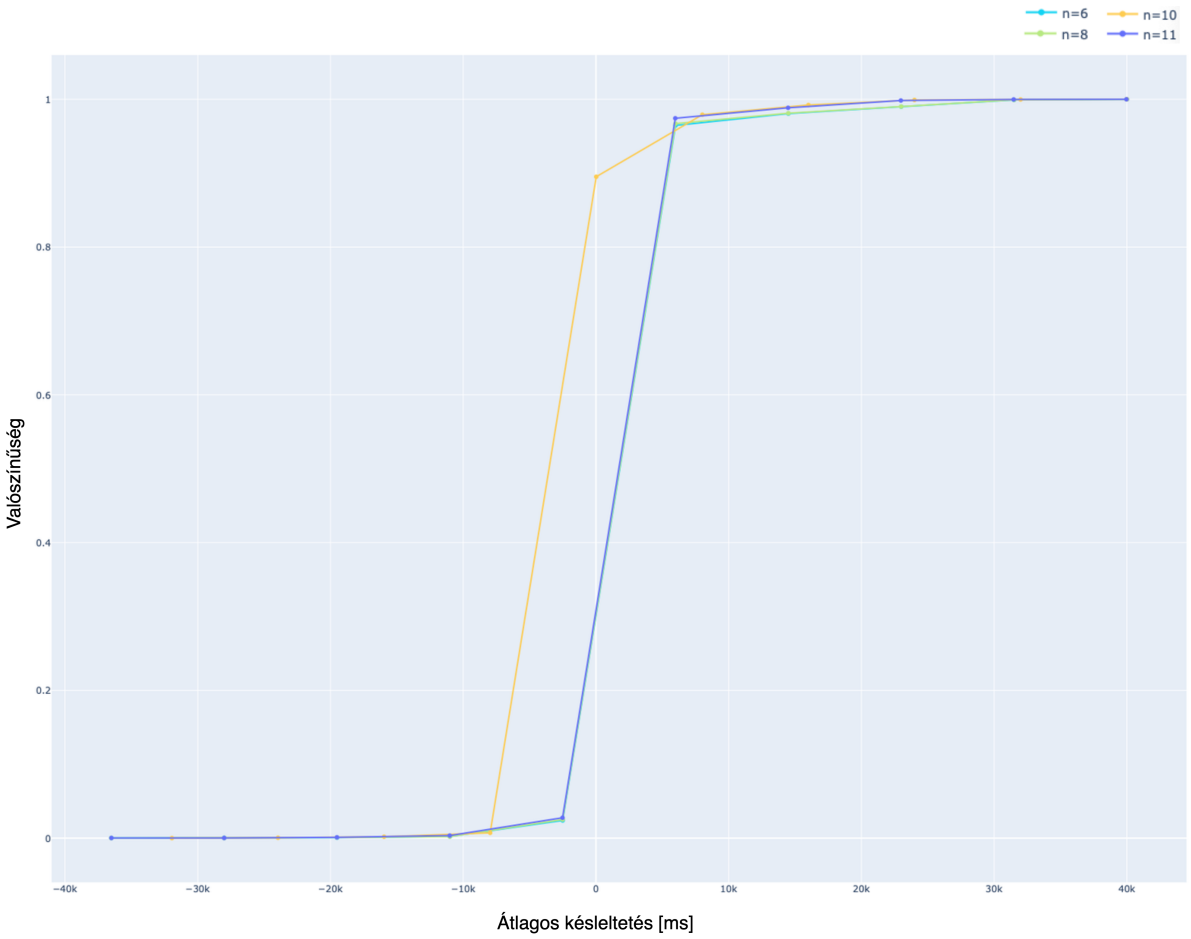


3.7. ábra. Az *n* paraméter függvényében bekövetkező csomagvesztés.

Az ábra alapján látható, hogy az első oszlopban szinte minden LAN oldali irányváltás esetén egy vágás történt ami miatt valós statisztika nem tudott előállni. A második oszlopban viszont már lenne relevanciája a képviselt értékeknek, hiszen itt már nem kerül elvágásra annyi csomag (egy TCP kapcsolat háromutas kézfogása már ekkor le tudott zajlani), viszont az analitikába nem vesszük figyelembe, hiszen legalább két csomagot érdemes ismerni ahhoz, hogy feldolgozható eredmény álljon elő az egyes halmazokra.

A vizsgálandó 5-11-es paraméterhez mindenképpen azt az ideális eseteket szükséges kiválogatni, ahol lehetőleg minimalizálva van a paraméter függvényében bekövetkező csomagvesztések száma, melyek a 6,8,10 és 11-es paraméter konfigurációk esete, hiszen ekkor látható a legkisebb megjelenő szummázott darabszám.

Ezen eseteknek fontos az eloszlásfüggvényét megvizsgálni, amelyen ábrázolva van a két halmaz késleltetésének relációja a 2.1. formula szerint, hiszen ez okvetlenül leírja mekkora valószínűséggel vesz fel ezen n konfigurációs paraméter mellett szolgáltatásromlás összegbe eső késleltetési értéket. Az egyes eloszlásfüggvények a 3.8. ábrán láthatóak, ahol a világoskék *(i)* az $n = 6$, *(ii)* a sárga $n = 10$, *(iii)* a világoszöld $n = 8$ és *(iv)* a sötétkék az $n = 11$-es paraméter konfiguráció görbéjét jelölik.



3.8. ábra. Konfiguráció paraméterek eloszlásfüggvénye.

Az x tengely mentén meghatározott átlagos késleltetés a megfigyelt és nem-megfigyelt álla-

potok által kalkulált differenciából adódik. A differencia előállítása a 0-ra való átrendezéséből következik a 2.1. eljárásnak. A görbék íve alapján megállapítható, hogy mely n paraméter konfiguráció mellett érhető el legnagyobb valószínűséggel az ideális állapot, vagyis a szolgáltatásromlás mentes kommunikáció. Az $n \in [6, 8, 11]$-es eset láthatóan jelentős különbözettel bír a $n = 10$-es paraméter konfigurációhoz képest, amely különbözet negatív értelemben értendő, így ezáltal a továbbiakban ezen paraméterértékkel nem szükséges a vizsgálatot folytatni. Ezen konfiguráció esetén a lehető legnagyobb valószínűséggel jelenik meg az az eset amikor a két állapot különbsége konvergál a 0 értékhez, vagyis ekkor a legnagyobb valószínűséggel írható le a teljes adatfolyam, ha ismerjük az első 10 csomagját független a kommunikáció irányultságától. Számszerűsítve ennek a valószínűsége 0.8880 (természetesen ezen vizsgált halmaz esetén meghatározott érték elképzelhető, hogy más adatforgalomtípusok mellett más paraméterkonfigurációhoz igazodnak).

3.3. Horizontális szolgáltatásromlás elemzése

A következőkben a 2.5.1. fejezetben bemutatott eljárás alkalmazása esetén a korábban megállapított $n = 10$ paraméterértékkel történő mérési eredmények elemzése olvasható. Az elemzés során a 3.2. fejezetben kielemzett négy fő konfiguráció mentén vizsgálja a dolgozat a különböző statisztikai metrikákat, ezen metrikák a nem-megfigyelt állapotra vonatkozóan a 3.3. táblázatban láthatóak. A számértékek az egyes diszjunk adatfolyamokra vonatkoznak és azokra összegezve, ahol a szolgáltatásromlás fellépett a korai áramlási jellemzők alapján alkalmazva a maximum likelihood becslést.

3.3. táblázat. Nem-megfigyelt állapotban észlelt szolgáltatásromlás előfordulási statisztikák.

Metrika	**n**			
	6	**8**	**11**	**10**
Összes eset	45 865	46 960	47 582	47 461
Átlag	3 820,6372	3 722,9214	3 556,7364	3 568,7076
Szórás	8 503,4254	8 397,5914	8 114,2669	8 136,5016
Első kvartilis	0,5	0,5	0,5	0,5
Második kvartilis	6	7	9,5	8,5
Harmadik kvartilis	208	207	233,6875	219,3333
Összáramlás százalékos aránya	0,4547	0,4655	0,4717	0,4705

A négy vizsgálandó konfiguráció esetében látható, hogy az egyes értékek nagyon közeli eredményeket mutatnak egymáshoz, így jelentős különbségeket nem lehet észlelni főként az $n = 11$ és $n = 10$ esetében. Itt a kettő konfigurációs érték között az összes detektált szolgáltatás romlás 121 adatfolymaban maximalizálódik a 11-es beállítás javára. A szórást tekintve is kisebb a kilengés az átlagos késleltetések közötti becslési értékek szerint a 11-es beállítás javára. Ezek az eltérések mivel önmagukban nem szignifikánsak (függetlenűl a 11-es paraméter konfiguráció javára), illetve a korábbi megállapítások alapján az $n = 10$ konfiguráció paramétert

tekinti a dolgozat továbbá is ideálisnak. Látható, hogy ezen beállítás mellett az összes adatfolyam 47,05%-ban detektált szolgáltatásromlást. Természetesen fontos lenne ismerni, hogy ezen szolgáltatásromlás milyen módon befolyásolta a tényleges felhasználói élményt, illetve mekkora idővolument ölelt fel.

A megfigyelt állapotra vonatkozó statisztikai jellemzők a 3.4. táblázatban láthatóak. Az eljárás által megállapításra került a szolgáltatásromlás azonosítása a megfigyelt állapotban a korai áramlási jellemzők alapján. Fontos, hogy a megfelelő konfiguráció paraméter függvényében a nem-megfigyelt ágon maximalizáljuk az összeáramlás százalékos arányát, míg a megfigyelt halmazra vetőlegesen pedig minimalizálni próbáljuk azt.

3.4. táblázat. Megfigyelt állapotban észlelt szolgáltatásromlás előfordulási statisztikák.

Metrika	**n**			
	6	**8**	**11**	**10**
Összes eset	30 921	28 120	22 628	25 326
Átlag	549,67023	369,8036	407,8254	372,6517
Szórás	2223,1618	1641,1716	1 777,2577	1 693,4799
Első kvartilis	0,3333	0,25	0,3333	0,3333
Második kvartilis	3,75	2,5	2,5	2,0
Harmadik kvartilis	41,0	31,6875	41,0	34,0
Összáramlás százalékos aránya	0,3065	0,2788	0,2243	0,2510

A statisztikai mutatók alapján látható, hogy az összáramlás százalékos arányait tekintve az esetek 25,10%-ban volt szolgáltatásromlás detektálás a megfigyelt állapottérben az eljárás alapján. A 11-es konfiguráció esetben látható ennél jobb eredmény (22,43%), hiszen a megfigyelt állapottérben cél az összeáramlás százalékos arányának minimalizálása, viszont ez a differencia csekély ezáltal jelentős különbség nem látható a két érték között. A 3.8. ábra alapján beazonosított magas valószínűséggel előálló zéró végeredmény, viszont indikálja az $n = 10$ választását.

3.4. Vertikális szolgáltatásromlás elemzése

Ezen szekcióban a 2.5.2. fejezetben bemutatott eljárás alkalmazása esetén előálló implikációkat tanulmányozza a dolgozat, ahol az adatfolyamok nem-megfigyelt állapotában fellépő szolgáltatásromlás által keletkezett negatív hatások kerülnek vizsgálatra más adatfolyamok megfigyelt állapotában.

A vertikális szolgáltatásromlás validálása során a vizsgált adahalmaz eseteben nem figyeltem meg szolgáltatásromlásra utaló jeleket. Ez azonban nem zarja ki egyértelműen, hogy a javasolt eljárás nem működik megfelelően. Hosszú távú vizsgálatokra van kétségtelenűl szükseg ahhoz, hogy végleges következtetéseket lehessen levonni a szolgáltatásromlás észlelése tekintetében, amely hálózati forgalom korai áramlási jellemzőinek felhasználásával történik.

A validálás során a javasolt eljárás nem talált vertikálisan azonosítható szolgáltatásromlást. Meg kell azonban jegyezni, hogy ez valoszínűleg a relative rövid időtartamnak tudható be,

hisz alig 300 másodpercnyi adatforgalmon történt az eljárás vertikális jellegű validálása. A forgalmi nyomvonal méretének növelésével mindenbizonnyal kedvezőbb adathalmaz állíthato elő. Ez azonban a PCAP fálj méretének rovására érhető csak el. A jelenlegi 300 másodperces forgalmi nyomvonal PCAP mérete közel 50 GB, amely lényeges korlátokat jelez előre nagyobb időtarmalmat magába foglaló forgalmi nyomvonalak tesztelésére.

(i) Egy masik megközelítésből, az a tény, hogy a vizsgált adathalmazban nem volt vertikálisan azonosítható szolgáltatásromlás annak is lehetett az eredménye, hogy a forgalmi nyomvonalban egyszerűen nem állt fenn semmilyen jellegű vertikális jelleget mutató szolgaltatásromlás. Kétségtelenűl, az ilyen jellegű vizsgálatok hosszútávú méréseket és elemzéseket igényelnek, amelyekre a munkám során nem volt lehetősegem.

(ii) Azt is jelentheti, hogy a szolgáltatásrolmás behatárolása, más megelőző folyamok alapján további finomításokat igényel. Hisz a javasolt eljáras szerint most csak azokat a szolgáltatásromlásokat vizsgáltam amelyek jól behatárolhatók, az átmeneteket és a küszöbértékeket el hanyagolva.

(iii) Alternatív megközelítésből, a nyomkövetést úgy is el lehetett volna végezni, hogy csak a csomagok fejlécei kerüljenek be a PCAP fájlba az adatrészek eldobása mellett. Ez azonban nem volt járható út, mivel az NFStream teljeskörű adatanalitikai működéséhez szükséges a csomagok adatrésze is, valamint kvantifikálni szeretném milyen applikáció típusra mekkora volument mutat a szolgáltatásrolás fellépése. Az adateldobás esetén több fejrész tudna egy PCAP fájlban megjelenni ami tárolás tekintetében hatékony, viszont a vizsgálati stádumban nem lehetne így kvantifikálni megfelelő pontossággal a hordozott informáicó alapján történő alkalmazáskategorizálást.

E korlátozás kiküszöbölése és a validálas kiterjesztett elvégzése érdekében, az ebben a munkabán bemutatott eljárást egy NFPlugin komponensen keresztüli implementálása javasolt. Az NFStream alapvető funkciói az NFPlugin bővítmény segítségével bővíthetők. Az NFPlugin általi implementálás lehetővé teszi az eljárás futtatását valós időben, reális környezetben, kiküszöbölve így a PCAP fájl magas tárolási igényeit. A forgalmi nyomkövetést úgy is el lehetett volna végezni, hogy csak a csomagok fejlécei kerüljenek be a PCAP fájlba az adatrészek eldobasa mellett. Ez azonban az NFStream teljeskörű adatanalitikai működését negatívan befolyásolja. További vizsgálatokra van szükség, hogy milyen kompromisszumok függvényében valosítható meg az ilyen jellegű forgalmi nyomkövetések feldolgozása NFStreammel.

Összegzés

A dolgozat keretén belül megvizsgáltam a szolgáltatásromlás észlelésének lehetőségét a hálózati forgalom korai áramlási jellemzőinek felhasználásával. A dolgozat rálátást ad arra, hogy az első 10 csomag korai áramlási jellemzőit megfigyelve 0.8880-os valószínűséggel tudja ideálisan lefedni a két állapot (megfigyelt és nem-megfigyelt) átlag késleltetés értékeinek diferenciáját, illetve az $n = 10$ konfigurációs paraméter minimalizálja az információs csomagvesztést és maximalizálja a két halmaz késleltetés számának különbségét. Rálátást biztosít a dolgozat, hogy az adott konfigurációs paraméter függvényében tényleges az összáramlás százalékos arányát tekintve 47,05%-ban realizál valamiféle szolgáltatásromlás jellegű karakterisztikát.

Az elemzésből kiderül, hogy vertikális szolgáltatásromlást nem sikerült észlelni az előálló adathalmazon, amelynek számos különböző forrása lehet. Bele értve a rövid monitorizált adathalmaz, illetve a kérdéses megléte a tényleges fellépésének a szolgáltatásromlásnak a halmazban. A dolgozat rámutat, milyen tényezői lehetnek ennek a problémakör nem megjelenésének, illetve mik azok az alternatív lépések amiket szükséges megtenni mélyrehatóbb elemzés elkészítéséhez. Továbbá javasolatot ad a dolgozat miként lehetne elvégezni a mérést és milyen szükségletek vannak az eljárás pontos validálásához. A jelen munka keretein túlmenően a kutatásban tett megállapítások iránymutatásként szolgálhatnak a gépi tanuláson alapuló alkalmazások osztályozási konfigurációjához is, különösen az áramlásokban megfigyelt első n adatmennyiségen alapuló időben történő osztályozás tekintetében.

A jövőbeni munkámban tervezem, hogy egy modellező alkalmazással támogatom meg a megoldást, amely lehetőséget biztosít az adatfolyamok csomagszintű elemzésére vizuálisan, illetve az egyes vertikális implikációk leképezésére. Ezentúl magát a megoldást szeretném kiterjeszteni a WAN oldali késleltetések elemzésére is, illetve a megállapított statisztikákra alacsony erőforrásigényű eljárást készíteni amivel adaptívan lehet a QoS-t kezelni a megfigyelések függvényében úgy, hogy a különböz alkalmazás kategóriák szerint taglalásra kerüljőn az eljárás működése. Szeretném ezen kívül kiterjeszteni más vizsgálati adathalmazokra is az elemzést, hogy megbizonyosodjak a megoldás kvalitásnak, illetve alkalmazhatóságának létéről.

Irodalomjegyzék

[1] Z. Aouini, A. Kortebi és Y. Ghamri-Doudane, „Traffic monitoring in home networks: Enhancing diagnosis and performance tracking”, *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015, 545–550. old.

[2] A. Kortebi, Z. Aouini, M. Juren és J. Pazdera, „Home Networks Traffic Monitoring Case Study: Anomaly Detection”, *2016 Global Information Infrastructure and Networking Symposium (GIIS)*, 2016, 1–6. old.

[3] P. Velan, „Improving network flow definition: Formalization and applicability”, *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018. ápr., 1–5. old.

[4] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto és A. Pras, „Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX”, *IEEE Communications Surveys Tutorials*, 16. évf., 4. sz., 2037–2064. old., 2014.

[5] M. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster és Y. Song, „FlowQoS: QoS for the rest of us”, *HotSDN 2014 - Proceedings of the ACM SIGCOMM 2014 Workshop on Hot Topics in Software Defined Networking*, 2014. aug.

[6] Z. Aouini és A. Pekar, „NFStream: A flexible network data analysis framework”, *Computer Networks*, 204. évf., 108719. old., 2022.

[7] L. Bernaille és R. Teixeira, „Early Recognition of Encrypted Applications”, *Passive and Active Network Measurement*, S. Uhlig, K. Papagiannaki és O. Bonaventure, szerk., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, 165–175. old.

[8] R. Bar - Yanai, M. Langberg, D. Peleg és L. Roditty, „Realtime Classification for Encrypted Traffic”, *Experimental Algorithms*, P. Festa, szerk., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, 373–385. old.

[9] A. Dainotti, A. Pescapè és C. Sansone, „Early Classification of Network Traffic through Multi-classification”, 6613. köt., 2011. ápr., 122–135. old.

[10] Y. Kumano, S. Ata, N. Nakamura, Y. Nakahira és I. Oka, „Towards real-time processing for application identification of encrypted traffic”, *2014 International Conference on Computing, Networking and Communications (ICNC)*, 2014, 136–140. old.

[11] C. V. Wright, F. Monrose és G. M. Masson, „On Inferring Application Protocol Behaviors in Encrypted Network Traffic”, *Journal of Machine Learning Research*, 7. évf., 100. sz., 2745–2769. old., 2006. cím: `http://jmlr.org/papers/v7/wright06a.html`.

[12] C. Bacquet, A. N. Zincir-Heywood és M. I. Heywood, „An Investigation of Multi-objective Genetic Algorithms for Encrypted Traffic Identification”, *Computational Intelligence in Security for Information Systems*, Á. Herrero, P. Gastaldo, R. Zunino és E. Corchado, szerk., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, 93–100. old.

[13] R. Alshammari és A. N. Zincir-Heywood, „Machine learning based encrypted traffic classification: Identifying SSH and Skype”, *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, 1–8. old.

[14] D. J. Arndt és A. N. Zincir-Heywood, „A Comparison of three machine learning techniques for encrypted network traffic analysis”, *2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2011, 107–114. old.

[15] Y. Okada, S. Ata, N. Nakamura, Y. Nakahira és I. Oka, „Application identification from encrypted traffic based on characteristic changes by encryption”, *2011 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, 2011, 1–6. old.

[16] C. Bacquet, A. N. Zincir-Heywood és M. I. Heywood, „Genetic optimization and hierarchical clustering applied to encrypted traffic identification”, *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 2011, 194–201. old.

[17] T. Bujlow, T. Riaz és J. M. Pedersen, „A method for classification of network traffic based on C5.0 Machine Learning Algorithm”, *2012 International Conference on Computing, Networking and Communications (ICNC)*, 2012, 237–241. old.

[18] C. Bovy, H. Mertodimedjo, H. Uijterwaal, P. Mieghem és G. Hooghiemstra, „Analysis of end-to-end delay measurements in Internet”, 2002. jan.

[19] J.-C. Bolot, „End-to-End Packet Delay and Loss Behavior in the Internet”, *Conference Proceedings on Communications Architectures, Protocols and Applications*, SIGCOMM ’93 sor., San Francisco, California, USA: Association for Computing Machinery, 1993, 289–298. old. cím: `https://doi.org/10.1145/166237.166265`.