

# Hurkolt UDP és QUIC alapú szállítási réteg-mechanizmusok forgalmi tulajdonságai

## Traffic properties of looped UDP and QUIC-based transport layer mechanisms

Dr. GÁL Zoltán<sup>1</sup>, Dr. TERDIK György<sup>1</sup>

<sup>1</sup>Debreceni Egyetem Informatikai Kar  
4028 Debrecen, Kassai út 26., Magyarország, <http://www.inf.unideb.hu>

### Abstract

*The study examines the operation and performance differences of TCP, UDP and QUIC protocols in the modern network environment. TCP provides reliable, connection-oriented communication, but latency and head-of-line blocking can hamper performance. UDP is faster but less reliable, while QUIC uses UDP to address TCP's limitations, reduce latency, support multiplexing, and connection migration between networks. This research summary analyzes the effectiveness of QUIC and UDP in a real network environment, paying particular attention to data flow control and traffic periodicity. In a real data traffic environment, in the case of different combinations of environmental parameters (garden size, segment size, transmission rate), a 10 MB file was copied between nodes using the UDP and QUIC protocols. In addition to the new delivery capabilities of the QUIC mechanism, we present the special characteristics of the periodicity of the arrival intervals of the regulated UDP traffic of the data stream.*

**Keywords:** datagram, high-speed network, TCP, UDP, QUIC.

### Kivonat

*A tanulmány a TCP, UDP és QUIC protokollok működését és teljesítménybeli különbségeit vizsgálja a modern hálózati környezetben. A TCP megbízható, kapcsolatorientált kommunikációt biztosít, ám a késleltetés és a "head-of-line" blokkolás hátráltathatja a teljesítményt. Az UDP gyorsabb, de kevésbé megbízható, míg a QUIC az UDP-t használva jól kezeli a TCP korlátait, csökkenti a késleltetést, támogatja a multiplexelést és a hálózatok közötti kapcsolat-migrációt. Jelen kutatási összefoglaló elemzi a QUIC és az UDP hatékonyságát valós hálózati környezetben, különös figyelmet fordítva az adatfolyam szabályozásra és a forgalom periodicitására. Valós adatforgalmi környezetben különböző környezeti paraméter (kertméret, szegmensméret, átviteli ráta) kombinációk esetén egy 10 MB méretű állomány ismételt átmásolása történt meg UDP, illetve QUIC protokollt alkalmazó csomópontok között. A QUIC mechanizmus új szállítási képességei mellett az adatfolyam szabályozott UDP forgalom beérkezési időközéi periodicitásának speciális jellemzőit mutatjuk be.*

**Kulcsszavak:** datagram, nagysebességű hálózat, TCP, UDP, QUIC, klaszter, OPTICS algoritmus.

## 1. BEVEZETÉS

A TCP (Transmission Control Protocol) és az UDP (User Datagram Protocol) a számítógépes hálózatok alapvető szállítási rétegbeli protokolljai, amelyek mindegyike más-más szerepet tölt be. A TCP megbízható, kapcsolatorientált kommunikációt biztosít. Hibaészlelés, csomagrend és újraküldési mechanizmusok révén nyújtja az adatok integritását, így olyan alkalmazásokhoz rendelik, ahol az adatok pontossága kritikus, például fájlátvitel, e-mailek és webböngészés. A TCP háromirányú kézfogást alkalmaz a kapcsolat létrehozásához, valamint az áramlásszabályozás és a torlódások elkerülésének kezeléséhez. Ezzel szemben az UDP kapcsolat nélküli, gyorsabb, kevésbé megbízható adatátvitelt kínál. Szegmenseket küld a kapcsolat felépítése, illetve a kézbesítés garanciája nélkül, így ideális olyan időérzékeny alkalmazásokhoz, mint a videó streaming, az online játékok és a VoIP (Voice over IP), ahol a sebesség a pontosságnál fontosabb. Mivel az UDP nem rendelkezik beépített hibajavítással és adatfolyam szabályozással, minimális többletterhelést (overhead) okoz. Mindkét

protokoll kiegyensúlyozza a megbízhatóság és a sebesség közötti kompromisszumot, és az alkalmazások széles skáláját támogatja az Interneten.

A QUIC (Quick UDP Internet Connections) szállítási mechanizmus bevezetése a hagyományos Internetes átviteli protokollok, különösen a TCP (Transmission Control Protocol) kritikus korlátait miatt került bevezetésre a modern webhasználat esetén. Bár a TCP megbízható és széles körben elterjedt, nehezen kezeli a késleltetés, a head-of-line blokkolás és a kapcsolatkezelés kihívásokat a mai dinamikus igények mellett. A TCP többlépcsős kézfogási folyamata, amely elengedhetetlen a biztonságos kapcsolatok létrehozásához, növeli a késleltetést, ami problémássá válik az alacsony késleltetésű interakciókat igénylő alkalmazásoknál, mint például a streaming és a valós idejű kommunikáció.

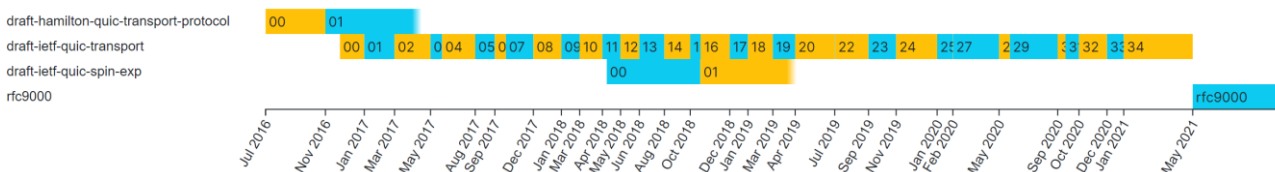
Ezen túlmenően, a TCP head-of-line blokkolása, ahol az egyik adatfolyam késése feltarthat másokat, hatással van a hatékonyságra több adatfolyam egyidejű továbbításakor. Ezáltal a TCP kevésbé optimális a több erőforrással rendelkező webalkalmazások vagy a csomagvesztéses esetekben. Mivel a TCP-kapcsolatok szorosan meghatározott IP-címekhez kapcsolódnak, komoly fennakadásokat okoz a felhasználók hálózat váltásakor, például Wi-Fi-ről mobilra eseten. Az UDP feletti működésre tervezett QUIC enyhíti ezeket a problémákat. Csökkenti a csatlakozási késleltetést azáltal, hogy egyetlen kézfogásban egyesíti a kapcsolatlétrehozást és a titkosítás beállítását, támogatja a multiplexelést, hogy elkerülje a head-of-line blokkolást, és lehetővé teszi a hálózatok közötti kapcsolat-migrációt megszakítás nélkül.

Ezek a funkciók a QUIC-ot jobban alkalmazkodóvá teszik a modern webes forgalom igényeihez, javítva a teljesítményt nagy késleltetésű, csomagvesztéses és gyakori hálózati változtatásokkal járó környezetben. A biztonságos adatátvitelt is leegyszerűsíti azáltal, hogy csökkenti a különálló titkosítási protokolloktól való függőséget. A QUIC iránti igény mára a gyorsabb, megbízhatóbb és biztonságosabb webes forgalomkezelés iránti növekvő keresletből adódik, ami robusztus megoldássá teszi a fejlődő Internet számára. Természetes kérdésként fogalmazódik meg, hogy ez az új mechanizmus számszerűsített módon mennyivel teljesít jobban, mint az ötven éve létező TCP, illetve UDP. Jelen dolgozatban erre keressük a választ és adunk iránymutatást.

A dolgozat további részének felépítése a következő: a QUIC mechanizmus működése és integrációja a hagyományos Internet protokoll veremszerkezetbe, a második részben található. Az UDP és a QUIC hatékonyságának elemzéséhez kapcsolódó munkák a harmadik részben található. Ezt követi mérési környezet és az elemzési eredmények bemutatása. Végül a következtetéseket vonjuk le és megadjuk a kapcsolódó kihívások kutatásának lehetséges folytatási irányát.

## 2. A QUIC SZÁLLÍTÁSI RÉTEG MEGHANIZMUS

A QUIC (Quick UDP Internet Connections) a modernebb szállítási rétegprotokoll, amely az UDP-n (User Datagram Protocol) keresztüli kapcsolatorientált webforgalom teljesítményének javítására szolgál. Eredetileg 2013-ban javasolta a Google, majd 2021-ben az IETF (Internet Engineering Task Force) szabványává vált RFC 9000 néven [1]. A QUIC elsődleges célja a széles körben használt TCP-ben (Transmission Control Protocol) rejlő korlátok kezelése, miközben javítja a biztonságot, az internet kapcsolatok sebességét és hatékonyságát.



1. ábra. QUIC mechanizmus szabványosítási folyamata 2024-ig

A QUIC-ot mára számos alkalmazásban használják, különösen a Google szolgáltatásaiban, mint például a YouTube, Meet és a Gmail, valamint más népszerű platformokon, például a Facebook/Messenger, WhatsApp, Uber, és mások. Integrációja a HTTP/3-ba, amely a HTTP protokoll jelenleg legújabb verziója, tovább erősíti pozícióját a jövő webes forgalom szállítási mechanizmusaként.

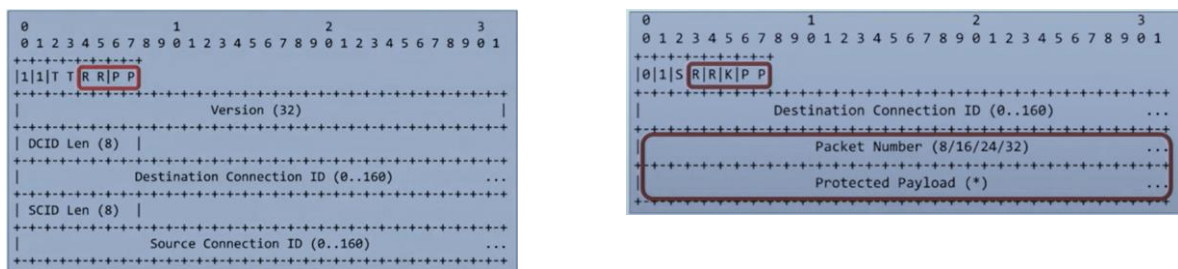
### 2.1. A QUIC főbb jellemzői

Az IP-rétegre épülő TCP-vel ellentétben a QUIC UDP felett működik, lehetővé téve, hogy megkerülje a TCP bizonyos korlátait, mint amilyen például a lassú kézfogást és a vonalak blokkolásának érzékenysége. Ez rugalmasságot biztosít olyan helyzetekben, ahol a TCP nehézségekbe ütközhet, például nagy késleltetésű hálózatok esetén. A QUIC-et úgy tervezték, hogy jelentősen csökkentse a kapcsolat létrehozásának idejét. A

TCP-vel ellentétben, amely többszörös oda-vissza utat igényel a kézfogáshoz és a titkosítás beállításához (TCP + TLS), a QUIC egyetlen kézfogásban egyesíti az átvitelt és a titkosítást. Ez szinte azonnali biztonságos kapcsolatokat tesz lehetővé, javítva a felhasználói élményt, különösen a mobil vagy valós idejű alkalmazásokban, ahol a várakozási idő kritikus.

A TCP egyik kritikus hátránya a head-of-line (HOL) blokkolás, ahol a csomagvesztés vagy késések egy adatfolyamban feltartják az összes többi adatfolyamot ugyanazon a kapcsolaton. A QUIC ezt úgy kezeli, hogy több adatfolyamot multiplexel egyetlen speciális logikai kapcsolaton belül. Ha az egyik adatfolyam csomagvesztést tapasztal, az nem érinti a többieket, ami hatékonyabb adatátvitelt és rövidebb letöltési időt eredményez a webhelyek számára. A biztonság a QUIC szerves része. Alapértelmezés szerint minden QUIC-kapcsolat titkosítva van, a titkosítási műveletekhez TLS 1.3-at használ. Azáltal, hogy a titkosítást közvetlenül a protokollba építi be, a QUIC megszünteti a külön titkosítási rétegek szükségességét (mint a TCP-n keresztüli TLS esetében). Ez biztonságosabbá teszi a QUIC-ot, nehezebben feltörhetővé és ellenállóbbá teszi a köztes hálózati támadásokkal szemben.

A QUIC másik innovatív funkciója a kapcsolat áttelepítése, dinamikus mozgatása. Mivel a QUIC-kapcsolatok nincsenek meghatározott IP-címekhez, például a TCP-portkhoz kötve, zökkenőmentesen migrálhatnak a hálózatok között (például Wi-Fi-ről mobilra váltva), anélkül, hogy újra fel kellene építeniük a kapcsolatot. Ez különösen hasznos azoknak a mobilfelhasználóknak, akik gyakran változtatják a hálózati interfészeket. A QUIC fejlett torlódás-ellenőrző algoritmusokkal rendelkezik, amelyek lehetővé teszik, hogy gyorsan alkalmazkodjon a változó hálózati feltételekhez. A TCP-vel ellentétben, amelyet a visszamenőleges kompatibilitás fenntartásának szükségessége korlátoz, az UDP feletti QUIC funkcionalitása bővíthet és hatékonyabb torlódás-ellenőrzési technikák alkalmazhatók, tovább javítva teljesítményét.



2. ábra. A QUIC LA adatelem szerkezete. Hosszú fejrész (bal), rövid fejrész (jobb).

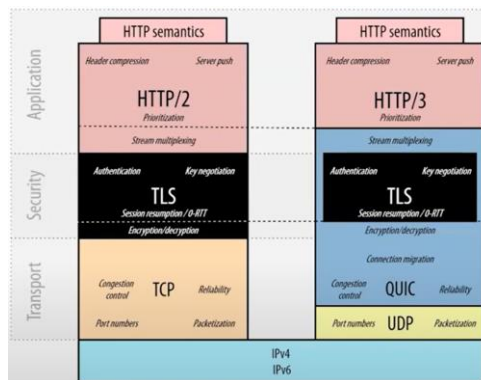
Míg a QUIC UDP felett működik, amely hagyományosan nem rendelkezik hibajavító vagy újraküldési mechanizmusokkal, a QUIC megvalósítja saját megbízhatósági funkcióit. Az elveszett csomagokat hatékonyan küldi újra anélkül, hogy a TCP-ben tapasztalt késéseket okozná, tovább csökkentve a HOL blokkolásának esélyét. A QUIC csökkentett kézfogási ideje és multiplexelési képességei jelentősen felgyorsítják a weboldalak betöltését, így különösen előnyös a sok erőforrással (képekkel, szkriptekkel, stb.) rendelkező webhelyek számára. A felhasználók stabil és gyors kapcsolatokat tarthatnak fenn megszakítás nélkül a hálózatváltások során. A QUIC azon képessége, hogy hatékonyabban tudja kezelni a csomagvesztést, alkalmassá teszi video- és hangátvitelre, ahol az adatok időben történő szállítása döntő fontosságú. A QUIC kötelező titkosítása megakadályozza, hogy harmadik fél elfogja vagy módosítsa az adatokat, így azok természetüknél fogva biztonságosabbak, mint a hagyományos TCP-kapcsolatok.

Meg kell említenünk, hogy előnyei ellenére a QUIC manapság néhány kihívással néz szembe. Egyes régebbi hálózati berendezéseket nem QUIC-forgalom kezelésére tervezték, ami interferenciához vagy teljesítményromláshoz vezethet bizonyos hálózatokon. Mivel a QUIC egyesíti a szállítási és kriptográfiai rétegeket, a protokollt bonyolultabb megvalósítani, mint a TCP-t.

## 2.2. A QUIC és a TLS kapcsolata

A QUIC és a TLS (Transport Layer Security) szorosan integrálódnak, mivel a QUIC a titkosításhoz és hitelesítéshez közvetlenül építi be a TLS-t, és egyszerűsíti a biztonságos Internetes kommunikációt. A QUIC alapfunkcióként írja elő a titkosítást, és a TLS 1.3-at használja kriptográfiai műveleteihez. Ezzel szükségtelenné válik külön titkosítási réteg (mint a TCP-ben, amely a felette TLS-re támaszkodik), biztosítva, hogy minden QUIC-kapcsolat automatikusan, további konfiguráció nélkül biztonságos legyen. A QUIC-kapcsolat létrehozásának pillanatától kezdve a titkosítás aktív, biztosítva az adatok integritását és hitelesítését. A hagyományos TCP-n keresztüli TLS-ben a kapcsolatlétesítés folyamata többszörös oda-vissza utat igényel (a TCP-kézfogást egy külön TLS-kézfogás követi). A QUIC ezt úgy optimalizálja, hogy a szállítási réteg

beállítását és a TLS-kézfogást egyetlen lépésbe végzi el. Ez drasztikusan csökkenti a biztonságos kapcsolat létrehozásához szükséges időt, hozzájárulva a QUIC gyors működéséhez.



3. ábra. A QUIC mechanizmus UDP, TLS és HTTP/3 kapcsolata

A QUIC támogatja a gyorsabb kapcsolatbeállítást a TLS 1.3 munkamenet-újraindítási funkciói révén. Amikor egy kliens újra csatlakozik egy kiszolgálóhoz, amellyel korábban kommunikált, a QUIC lehetővé teszi a munkamenet-újraindítási kulcsok használatát, csökkentve a teljes kézfogás szükségességét, és javítva a kapcsolatok helyreállításának sebességét. A QUIC a TLS 1.3-at közvetlenül a protokollba integrálja, biztosítva, hogy minden kapcsolat a kezdetektől fogva biztonságos legyen. A beintegrált TLS csak a titkosítási, integritási és hitelesítési szempontokért felelős, míg a szállítási funkciókat, például a torlódáskezelést és az újraküldést a QUIC más modulja kezeli. Ez az egyértelmű szétválasztás lehetővé teszi a QUIC számára, hogy rugalmas szállítási mechanizmusokat valósítson meg, miközben a TLS 1.3 robusztus szolgáltatásai révén megőrzi a biztonságot. A szállítási és biztonsági mechanizmusok ezen kombinációja növeli a teljesítményt és a biztonságot, csökkenti a csatlakozási időket és egyszerűsíti a biztonságos kommunikációt az Interneten keresztül.

### 2.3. A QUIC és a HTTP kapcsolata

A QUIC és a HTTP protokoll alapvetően összefügg egymással, különösen a HTTP/3 bevezetésével, amely a QUIC szállítási mechanizmusának kihasználására épül. Hagyományosan a HTTP TCP-re támaszkodott az adatátvitelhez (HTTP/1.1 és HTTP/2), de a QUIC a TCP-t váltja fel a HTTP/3 szállítási rétegeként a teljesítmény- és hatékonysági kihívások kezelésére. A TCP, bár megbízható, késéseket vezet be a többlépcsős kézfogás, az újraküldési stratégiák és a head-of-line (HOL) blokkolás miatt. A QUIC az UDP-n keresztüli működésével csökkenti ezeket a késéseket a szállítási és biztonsági rétegek kombinálásával, lehetővé téve a gyorsabb kapcsolatépítést és hatékonyabb adatkezelést.

A HTTP/2 bevezette a multiplexelést, amely több kérést és választ tesz lehetővé egyetlen TCP-kapcsolaton keresztül. A HTTP/2 azonban továbbra is HOL blokkolást szenved a TCP rétegen. A QUIC belső kialakítása támogatja a valódi multiplexelést azáltal, hogy megakadályozza, hogy az egyik adatfolyamban lévő csomagvesztés másokat érintsen. Ez javítja a teljesítményt számos erőforrást (például képeket, szkripteket és stíluslapokat) tartalmazó weboldalak betöltésekor. A modern webes környezetekben a felhasználók gyakran váltanak a hálózatok között (pl. Wi-Fi-ről mobil adatátvitelre). A QUIC lehetővé teszi a zökkenőmentes kapcsolatáttelepítést, ami azt jelenti, hogy a HTTP/3 munkamenetek megszakítás nélkül folytatódhatnak, simább böngészési élményt biztosítva. Ezzel szemben a TCP-kapcsolatokat hálózati váltás után újra kell építeni.

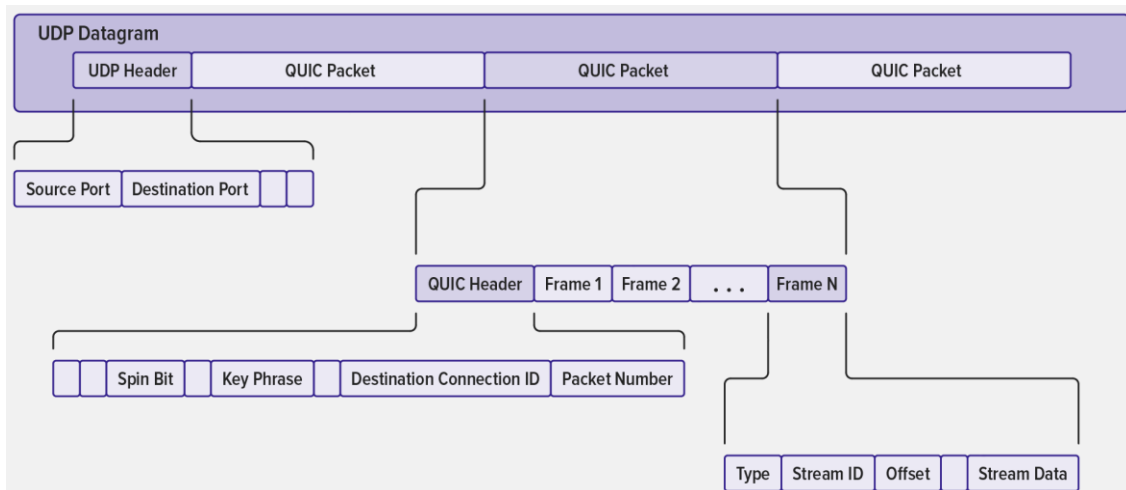
A QUIC integrált titkosítása biztosítja, hogy a HTTP/3 forgalom mindig titkosítva legyen, csökkentve a többletterhelést, valamint gyorsabbá és hatékonyabbá téve a biztonságos kapcsolatokat. A QUIC HTTP/3-mal való integrálásának egyik fő célja a késleltetés csökkentése. A TCP által megkövetelt többszöri oda-vissza kézfogás kiküszöbölésével a QUIC gyorsabb oldalbetöltést és alacsonyabb várakozási időt tesz lehetővé, ami különösen fontos a mobil és a valós idejű alkalmazások, például a videó streaming vagy a játékok esetében.

A QUIC bevezetésével a HTTP/3 hatékonyabb, biztonságosabb és gyorsabb adatátviteli folyamatot kínál, optimalizálva a web működését, különösen a késleltetésre érzékeny környezetekben. Az integráció megoldja a HTTP korábbi verzióinak kulcsfontosságú problémáit, így jobban megfelel a modern adathálózati igényeknek.



## 2.4. A QUIC adatfolyam szabályozó mechanizmusa

A QUIC egy kifinomult áramlásszabályozó rendszert valósít meg, amely mind a csatlakozási, mind az adatfolyam szinten működik. Ez lehetővé teszi, hogy a protokoll kezelje az adatáramlást, miközben elkerüli a puffer túlsordulást, és biztosítja a több adatfolyam közötti méltányosságot (fairness). A továbbiakban röviden soroljuk fel e mechanizmus jellemzőit.



4. ábra. A QUIC, illetve az UDP protokoll adatainak viszonya

a) *Adatfolyam-szintű áramlásszabályozás:* A QUIC lehetővé teszi az adatok több független adatfolyamon történő továbbítását egyetlen kapcsolatban. Minden adatfolyamnak megvan a saját áramlásvezérlő mechanizmusa, amelyet a vevő irányít, és amely meghatározza az adott adatfolyamon nyugtázás nélkül küldhető maximális adatmennyiséget. A vevő egy adatfolyam-vezérlő ablakot kommunikál, amely nem más, mint az az adatmennyiség, amelyet a küldő tud továbbítani anélkül, hogy túlterhelné a vevőt. A küldőnek tiszteletben kell tartania ezt az ablakot, és meg kell várnia a fogadó frissítéseit, hogy további adatokat küldhessen. Az egyes adatfolyamok független áramlásvezérlésével a QUIC elkerüli a vonalak blokkolását, ahol az egyik adatfolyam torlódása késleltetné a többiekét. Mindegyik adatfolyam halad a saját továbbításával anélkül, hogy befolyásolná a többiekét.

b) *Csatlakozási szintű áramlásszabályozás:* Az egyes adatfolyamok áramlásszabályozásán túlmenően a QUIC kapcsolati szinten is áramlásszabályozást alkalmaz. Ez biztosítja, hogy az összes adatfolyamon elküldött adatok teljes mennyisége ne haladja meg a vevő által kezelhető mennyiséget. A kapcsolatfolyamat-vezérlő ablak felső korlátot állít be az összes adatfolyamon futás közbeni bajtok teljes számára. A stream-szintű vezérléshez hasonlóan a vevő hirdeti ezt a korlátot, és a küldőnek úgy kell kezelnie az adásokat, hogy azok az engedélyezett ablakon belül maradjon.

c) *Ablakfrissítések:* Mind az adatfolyam-, mind a kapcsolatszintű áramlásvezérlő ablakok dinamikusan frissülnek. Ahogy a fogadó feldolgozza az adatokat, ablakfrissítéseket küld a feladónak, növelve a továbbítható adatok megengedett mennyiségét. Ezek a frissítések segítenek fenntartani az egyensúlyt a küldő átviteli képessége és a fogadó képessége között a bejövő adatok feldolgozására, megakadályozva a puffer túlsordulását vagy a kapcsolat kimerülését.

d) *Torlódásgátló integráció:* A QUIC áramlásvezérlése a torlódás-ellenőrző mechanizmusokkal együtt működik, amelyek a hálózat túlterhelésének megelőzése érdekében léteznek. Az áramlásszabályozás gondoskodik arról, hogy a vevő ne legyen túlterhelve, míg a torlódásvezérlés gondoskodik arról, hogy maga a hálózat ne legyen túlterhelt. A QUIC támogatja az olyan torlódáscsökkentő algoritmusokat, mint a NewReno vagy a Cubic, amelyek a küldési sebességet a hálózati feltételek, például a csomagvesztés vagy a késleltetés alapján igazítják.

e) *Adatok nyugtázása (ACK):* Megbízható átviteli rendszerének részeként a QUIC nyugtákat (ACK) használ az adatok fogadásának jelzésére. A vevő ACK-eket küld, hogy tájékoztassa a feladót bizonyos csomagok sikeres vételéről, lehetővé téve a küldő számára, hogy további adatot továbbítson. Ezek az ACK nyugták az áramlásszabályozó ablakok dinamikus beállításához is kulcsfontosságúak. Ha a küldő ésszerű időn belül nem kap ACK-t, akkor újraküldési mechanizmusokat indít el, miközben frissíti az áramlásvezérlést és a torlódást szabályozó algoritmusokat is.

Az adatfolyam vezérlés négy lépése a következő:

- 1) *A vevő beállítja a kezdeti korlátokat.* Ha QUIC-kapcsolat jön létre, a vevő beállítja a kezdeti áramlásszabályozási korlátokat mind az egyes adatfolyamokra, mind a kapcsolat egészére vonatkozóan. Ez tájékoztatja a küldőt arról, hogy mennyi adatot lehet továbbítani anélkül, hogy túlterhelné a vevőt.
- 2) *A küldő tiszteletben tartja az áramlásszabályozási ablakokat.* A küldő továbbítja az adatokat, de gondoskodik arról, hogy azok ne lépjék túl a fogadó által megszabott áramlásszabályozási határokat. A küldő nyomon követi, hogy mennyi adatot küldött, és várja a frissítéseket a fogadótól.
- 3) *A fogadó ablakfrissítéseket küld.* Ahogy a fogadó feldolgozza az adatokat, az áramlásvezérlő ablak frissítéseit küldi a feladónak, lehetővé téve több adat továbbítását. Ezek a frissítések gyakran megtörténhetnek, biztosítva az állandó adatáramlást anélkül, hogy torlódást vagy puffertúlcsordulást okoznának.
- 4) *Dinamikus hangolás.* Mind a küldő, mind a vevő dinamikusan állítja be az áramlásszabályozási korlátokat a hálózati feltételek, az alkalmazási igények és a torlódáskezelési algoritmusok alapján. Ez a dinamikus beállítás hatékony adatátvitelt biztosít, miközben megakadályozza a hálózati torlódást vagy a vevő túlterhelését.

Az áramlásvezérlés és a torlódásvezérlés kombinációja biztosítja, hogy a QUIC gyors, megbízható és megfelelően nagy adatátvitelt biztosítson, még összetett vagy változó hálózati körülmények között is.

### 3. AZ UDP, ILLETVE QUIC MECHANIZMUSOK TELJESÍTMÉNYÉT ELEMZŐ FONTOSABB MUNKÁK ÁTTEKINTÉSE

Az UDP egyszerűsége miatt ritkán kerül közvetlenül tudományos elemzések fókuszába. Ezzel szemben sok tanulmány foglalkozik a TCP és UDP összehasonlításával [16]. Az UDP hatékonyabb rövid fájlok esetén, míg a TCP csomagvesztés kompenzálása miatt hosszabb fájlok átvitelénél előnyösebb. Erős hálózati terhelésnél a TCP magas késleltetést okozhat, míg az UDP esetében adatvesztés történhet. A két protokoll eltérő QoS-szolgáltatást nyújt, amely az adott hálózati alkalmazástól függően változik.

Az átviteli teljesítmény, késleltetés és csomagvesztés szerinti összehasonlításokat ns-2 szimulátorral is elvégezték [5], ahol a hálózati terhelés és a topológia alapján különböztek az eredmények. A TCP hatékonyságát befolyásolja a torlódáskezelési algoritmus, míg mindkét protokoll esetében a puffer mérete is lényeges. Az algoritmusok különböző forgatókönyvekben eltérően viselkednek.

IEEE 802.11-es vezeték nélküli környezetben VoIP-forgalmat teszteltek UDP felett [10], elemezve a rádiós csatorna, csomagméret és csatorna minőségének hatását. Az UDP teljesítménye erősen függött a hálózat minőségétől, míg a VoIP stabil maradt. Hosszabb rádiós távolságoknál az UDP teljesítménye csökkent. Nagysebességű hálózatokban a PA-UDP protokollt javasolták [7], amely a hálózati körülményekhez igazodva dinamikusan változtatja a csomagméretet és az átviteli rátát. Magas terhelésnél a PA-UDP jobb teljesítményt nyújtott, mint a TCP vagy az UDP.

Egy másik javaslat, az UDT [14], amely nagy kiterjedésű hálózatokhoz készült, TCP-korlátokat pótol. Az UDT magas sebességű átvitelre alkalmas torlódásvezérléssel és adaptív adatfolyam-szabályozással. Visszacsatolási mechanizmust alkalmaz, amely a fogadó hálózati állapotától függően állítja a küldési rátát. Az RBUDP protokoll [15] szintén nagyméretű hálózatokra készült, nagysebességű fájlatvitelre. Adaptív torlódásvezérlést alkalmaz, és a csomag-fluxus technika révén több csomagot küld nyugtákozása nélkül.

A QUIC protokoll biztonságát és teljesítményét a [20] elemzés vizsgálta, amely mélyrehatóan foglalkozik titkosítással, multiplexeléssel és torlódásvezérléssel. A QUIC jobban teljesített a TCP-nél és TLS-nél, stabilabb biztonsági képességekkel. Azonban bizonyos esetekben nem garantálja az elfogulatlan kommunikációt, így egyes forgalmakat előnyben részesít másokkal szemben.

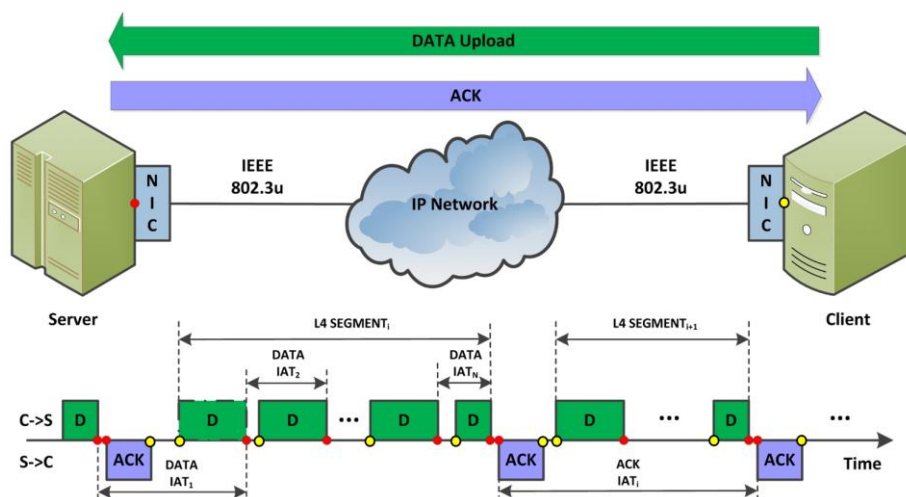
Visszacsatolós UDP forgalom esetén három független kommunikációs paraméter hatását elemezték: a maximális átviteli egység (MTU) mérete, a végpontok közötti munkamenet sáv szélessége és az UDP szegmens mérete [22]. A nem stacionárius többfelbontású módszerek alkalmazása lehetővé tette három jellegzetes forgalom-mintázat megkülönböztetését, amely a visszacsatolós datagram kommunikációs szolgáltatások objektív minőségi jellemzőinek beazonosítását támogatja.

Jelen dolgozat újszerűsége abban áll, hogy elemzési módszerekkel tárgyalja a datagram kommunikációs veremszerkezet feletti visszacsatolós, illetve adatfolyam szabályozás forgalmi aspektusait. Megmutatjuk a QUIC protokoll által generált beérkezési időközök függését az UDP szegmens mérete függvényében, valamint a valós hálózati környezetben mért forgalom esetén tapasztalt események periodicitásának változását.

## 4. HURKOLT UDP, ILLETVE QUIC FORGALMAK MÉRÉSE ÉS ELEMZÉSE

A klasszikus statisztikai időszerelemzés módszerei alkalmasak az adatkommunikációs szolgáltatások jellemzőinek értékelésére, különös tekintettel az adatok helyhez kötött sajátosságaira. Azonban a valós adatátvitel során mért idősorok gyakran nem-stacionáriusak, amit a hálózati igények gyors és kiszámíthatatlan változásai okoznak. Ezt a jelenséget több tényező befolyásolja, például a hálózati torlódások, a változó forgalmi terhelések, valamint a topológia dinamikus változásai. Nagysebességű hálózatokban a forgalmi volumen drámai ingadozásokat mutathat rövid idő alatt, ami a torlódások és a csomagvesztési arány növekedéséhez vezet. A hálózati topológia állapota, illetve az egyes csomópontok és kapcsolatok terhelése szintén változhat, ami az útválasztási útvonalak módosulását és a késleltetés növekedését eredményezi.

Ezen túlmenően a nagysebességű hálózatok gyakran összetett protokollokat és mechanizmusokat alkalmaznak, mint például a torlódásszabályozás és az adatfolyam-vezérlés, hogy az adatok időbeni és biztonságos kézbesítését biztosítsák. Ezek az algoritmusok hozzájárulnak a hálózati dinamika nem-stacionaritásához. A nagysebességű hálózatok erőteljesen nem-lineáris rendszerek, mivel az adattovábbítási képességük nem áll egyenes arányban a bemeneti terheléssel. A hálózat nem-lineáris viselkedése azt jelenti, hogy a bemeneti forgalom kis változásai jelentős hatást gyakorolhatnak a hálózat teljesítményére. A nem-linearitás egyik fő forrása a torlódás, ami csomagvesztéshez vagy késleltetéshez vezethet, így erőteljesen nem-lineáris viselkedést idézve elő. Továbbá, a hálózati protokollok komplexitása, különösen az erőforrás-korlátok miatt szükséges áramlásszabályozás, további nem-linearitást okoz.



5. ábra. Visszacatolós UDP, illetve QUIC alapú forgalom mérési környezete

Jelen kutatási munka során hurkolt UDP, illetve QUIC mechanizmus által továbbított fájlátvitelt mértünk és elemeztünk (ld. 5. ábra). Egy kliens gépről ugyanazt a 10 MB méretű fájlt különböző környezeti feltételek mellett másoltuk át (upload) ismételt formában. A független paraméter-hármas elemei a következők:

- a kliens gép hálózati kártyájának maximális keretmérete (MTU – Maximum Transfer Unit);
- az alkalmazás szintű adatátviteli sebesség (Bw – Bandwidth) viszonylagos értéke a csatornához;
- az UDP szegmens mérete (SSize – Segment Size).

Az UDP visszacsatolás a fogadó oldalon úgy történt, hogy a szerver nyugtát küldött vissza a kliensnek az érkező adatszegmensek alapján. Ez a nyugta két mezőből álló állapotinformációt tartalmazott: a) az UDP szegmenshez tartozó IP csomagok sikeres vagy sikertelen átvétele a fogadó szerver oldalán bináris jelöléssel (0 – IP csomag nem érkezett meg, 1 – IP csomag megérkezett); b) a nyugta küldésének időpontja a szerver oldalán. Az állapotinformáció a hiányzó IP csomagok újraküldéséhez szükséges, míg az időbélyeg az alkalmazás szintű adatátviteli sebesség figyelésére és szabályozására szolgál. A független hálózati paraméterek lehetséges értékei pedig a következők:

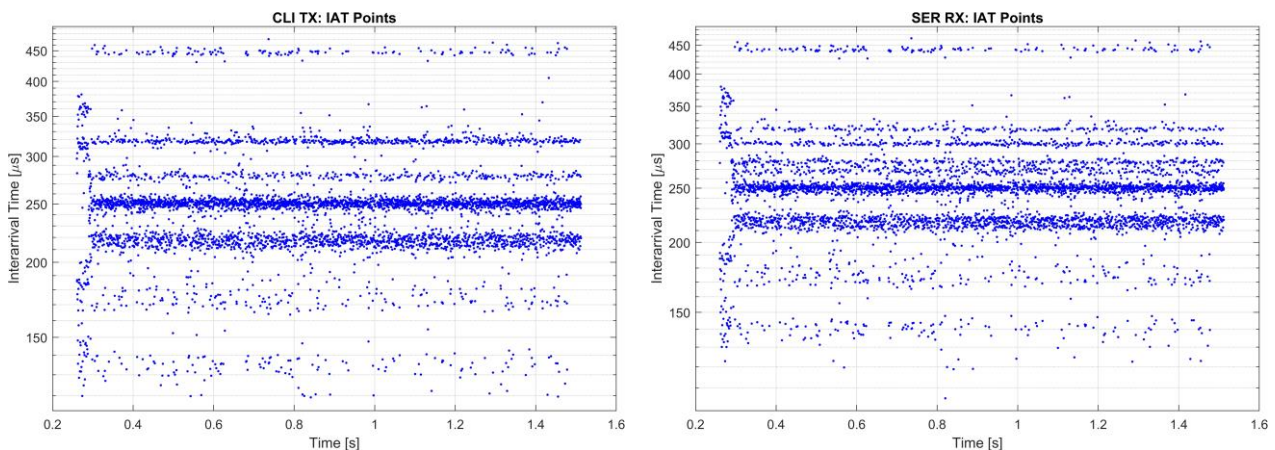
$$MTU[i] = 1260 + i \cdot 40 [B], \quad i = 1, 2, \dots, 6 \quad (1)$$

$$Bw[j] = 37 + j \cdot 3 [\%], \quad j = 1, 2, \dots, 21 \quad (2)$$

$$SSize[k] = 1 + k \cdot 3 [kB], \quad k = 1, 2, \dots, 19. \quad (3)$$

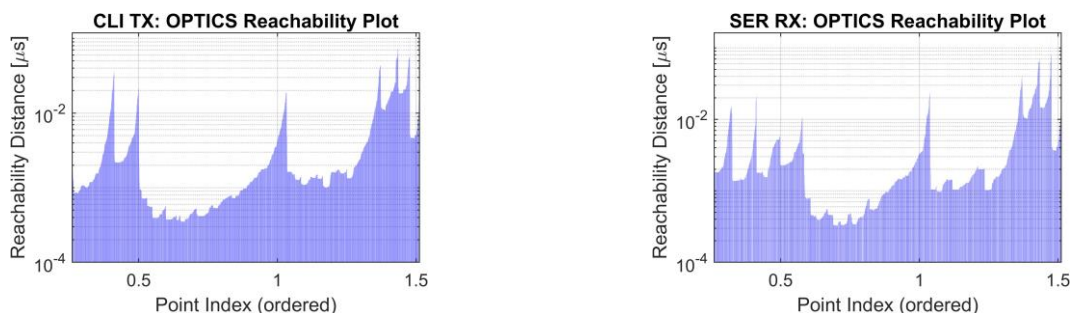
Ezáltal a maximális keretméret 1300 és 1500 bájt között, az alkalmazás szintű adatátviteli sebesség 40 és 100% között, az UDP szegmens pedig 4 és 58 kB közötti értékeket vett fel a képletek szerinti lépésközökkel. A [22] dolgozattal ellentétben nem csak a szerveren, hanem a kliensen is mértük mindkét irányba a továbbított Ethernet keretek időbélyegét, illetve méretét. Az összesen 2396 másolás során a mérési idő pontossága  $1 \mu\text{s}$ , a keretek hossza pedig bájtok egész számú többszöröse.

Az adatelemzés azt mutatja, hogy interfész kártya szinten periodikus események következnek be bizonyos időskálákon (ld. 6. ábra). Ez nem csak az adatkeret, IP csomag, illetve UDP szegmens, hanem a QUIC mechanizmus belső adatfolyam szabályozó képessége alapján történik. A küldő kliens és a vevő szerver oldalon mintavételezett IEEE 802.3 (Ethernet) keretek beérkezési időközét IAT (Interarrival Time) néven kezeljük a továbbiakban. Az IAT úgy a küldő, mint a vevő interfész kártyán jellemző főperiódusokat mutat, amelyek körül sztochasztikus jelleggel további értékek is megjelennek (ld. 6. ábra:bal és 6. ábra:jobb). Ezen tömörülések nem állandó szórással rendelkeznek még egyetlen mérésen belüli időser esetében sem.



6. ábra. Az IAT időközök klasztereződése,  $(MTU, Bw, SSize) = (1500 \text{ B}, 46 \%, 7 \text{ kB})$ ; Kliens küldési oldal (bal), Szerver fogadási oldal (jobb).

Az IAT értékek tömörülése lehetővé teszi, hogy megfelelő klaszterezési algoritmus segítségével külön válogassuk az egyes IAT csoportokat. A nem rögzített klasztersugár miatt OPTICS algoritmust használtunk fel, amely dinamikusan kezeli a klaszterek sugarát, azaz képes különböző sűrűségű csoportok kialakítására is. Az OPTICS klaszterezési algoritmus CD (Core Distance) és RD (Reachability Distance) metrikák segítségével az egymáshoz közeli pontok indexeit egy új sorrendbe helyezi el. Az RD metrika szomszédos csúcsai közötti elemek azonos klaszterbe tartoznak. A 7. ábra bal, illetve jobb oldalán például az  $(MTU, Bw, SSize) = (1500 \text{ B}, 46 \%, 7 \text{ kB})$  paraméterkombináció esetén megfigyelhető, hogy a kliens küldő, valamint szerver vevő idősorok RD oszlopdiagramjain hét, illetve nyolc „völgy” figyelhető meg, így ezen IAT idősorok hét, illetve nyolc klaszterbe sorolódnak.

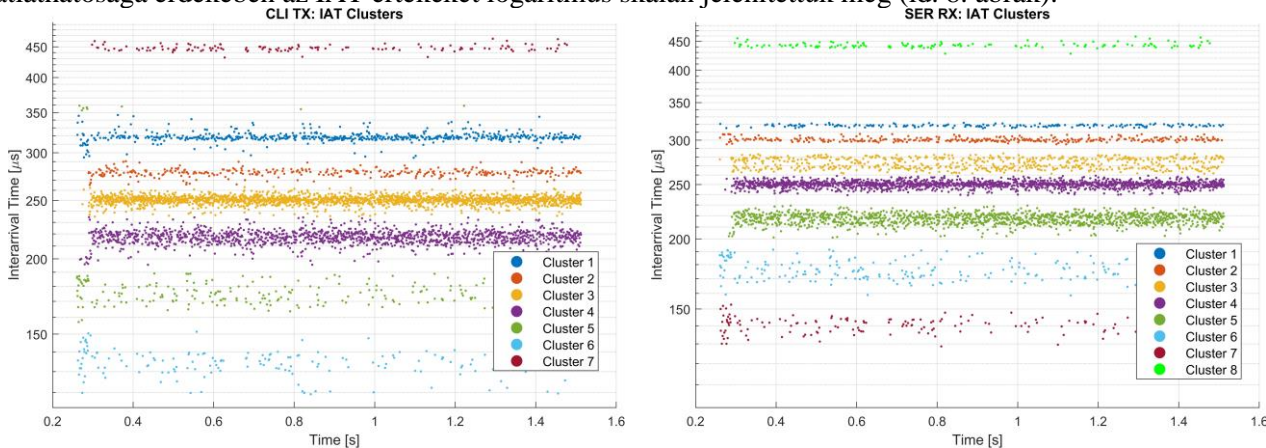


7. ábra. OPTICS algoritmus RD metrikája az IAT időközök klasztereződése esetén,  $(MTU, Bw, SSize) = (1500 \text{ B}, 46 \%, 7 \text{ kB})$ ; Kliens küldési oldal (bal), Szerver fogadási oldal (jobb).

Az így beazonosított klaszterek, azaz IAT időértékek a küldő kliens és a vevő szerver oldalon a 8. ábra szerint jeleníthetők meg. Fontos megjegyeznünk, hogy a klaszterszám változása küldő és vevő oldalon abból származik, hogy a továbbító lokális hálózati infrastruktúra (kapcsolók, átvásztók, stb.) a keret átvitele során tovább növelik az IAT értékek szórását. Mivel az OPTICS algoritmus adott távolság feletti tömörüléseket

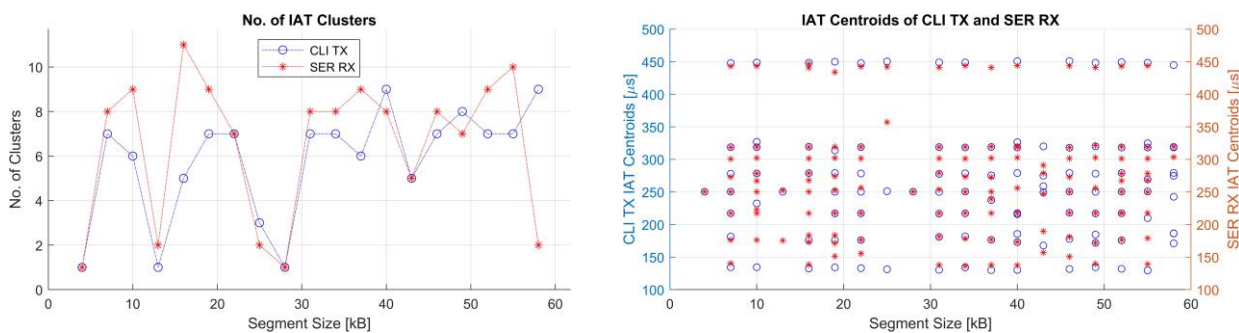


külön klaszterbe sorolja, ezért a küldő és a vevő oldalon eltérő klaszterszám adódik. Az időskálák jobb átláthatósága érdekében az IAT értékeket logaritmus skálán jelenítettük meg (ld. 8. ábrák).



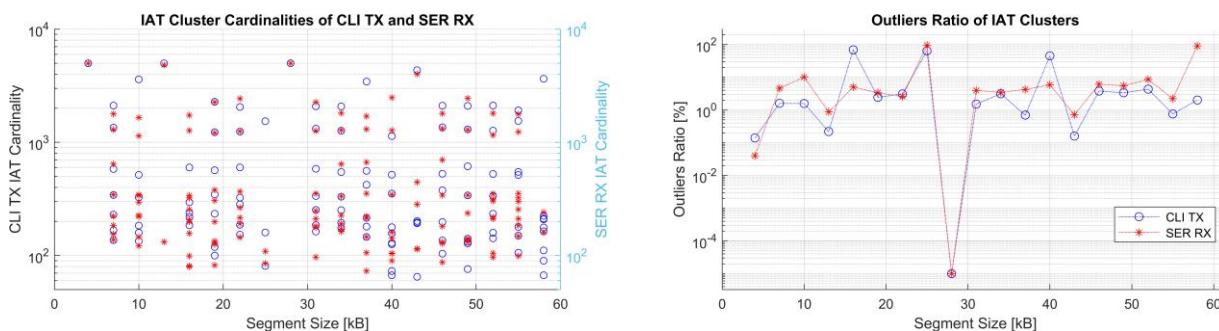
8. ábra. IAT időközök klasztereződése,  $(MTU, Bw, SSize) = (1500 B, 46 \%, 7 kB)$ .  
Kliens küldési oldal (bal); Szerver fogadási oldal (jobb).

Megvizsgáltuk a klaszterek vagyis az IAT főperiódusok darabszámának függését a szállítási réteg szegmenseinek mérete (SSize) alapján. Az esetek nagy többségében a szerver oldali fogadó interfész kártyán több klaszter jelenik meg, mint a kliens oldali küldőnél (ld. 9. ábra:bal). Kivételt képez a 27 kB-os szegmens méret, ahol egyetlen klaszter létezik mindkét oldalon. Ez azt jelenti, hogy a 27 kB-os szegmensméretet a QUIC optimális módon kezeli, az elküldött és megérkező Ethernet keretek egyetlen főperiódus szerint ismétlődnek a hálózatban.



9. ábra. IAT időközök klasztereződése,  $(MTU, Bw, SSize) = (1500 B, 46 \%, 4:3:58 kB)$ .  
Főperiódusok száma (bal); Főperiódusok értéke (jobb).

Nem minden egyes Ethernet keret küldése, vagy fogadása történik adott főperiódus környezetében. Léteznek ún. kirívó időközzel kezelt adatelemek is. Ezeket az OPTICS nem tudja klaszterbe sorolni, így statisztikai szempontból kiugró („outlier”) értékként kezeljük. Az egyes klaszterekbe tartozó Ethernet keretek darabszáma a 10. ábra:bal, míg a kiugró keretek százalékos aránya a 10. ábra:jobb oldalon látható. Itt is megfigyelhető, hogy a 27 kB-os szegmensméret speciális üzemmód mellett kerül továbbításra.



10. ábra. IAT időközök szignifikáns értékei, illetve a zaj,  $(MTU, Bw, SSize) = (1500 B, 46 \%, 4:3:58 kB)$ .  
IAT Klaszterek elemeinek száma (bal); IAT zaj százalékos aránya (jobb).

Mindezen események fizikai magyarázata a QUIC algoritmus adatfolyam szabályozó funkciójának belső tulajdonságaira vezethető vissza. Mivel a QUIC mechanizmus döntését a hálózati átvitel paramétereinek függvényében hozza meg, ezek időbeli módosulásának hatása dinamikus üzemmód módosulást okoz az algoritmus részéről.

## 5. ÖSSZEFOGLALÁS

A klasszikus statisztikai időszorelemzés eszközei jól használhatók az adatkommunikációs rendszerek jellemzésére, azonban a valós hálózati forgalom gyakran nem-stacionárius. A hálózati torlódások, változó terhelések és a dinamikus topológiai változások miatt a nagysebességű hálózatok nem-lineárisak, ahol a bemeneti forgalom kis változásai nagy teljesítményingadozást okozhatnak. A torlódás- és adatfolyam-vezérlési algoritmusok tovább növelik a hálózatok nem-linearitását. A QUIC jelentős előrelépést jelent a szállítási protokollok tervezésében, jelentős javulást kínálva a sebesség, a megbízhatóság és a biztonság terén. Ahogy egyre szélesebb körben alkalmazzák, különösen a HTTP/3 térnyerésével, az körvonalazódik, hogy lényegesen átalakítja az adatok interneten keresztüli továbbítását, optimalizálva a felhasználói élményt és a hálózati teljesítményt. Kutatásunk során UDP és QUIC alapú fájlátvitelt elemeztünk különböző hálózati paraméterek mellett. Az OPTICS algoritmus segítségével az IAT klaszterek változásait vizsgáltuk, és kiemeltük a QUIC algoritmus adatfolyam-szabályozásának szerepét. Az adatok azt mutatják, hogy az Ethernet keretek beérkezési időközei (IAT) klaszterezhetők. Az optimális szegmensméret 27 kB, amely minimalizálja a forgalmi periodicitást, illetve az IAT értékek szórását. További kutatások, illetve nemlineáris és nemstacionárius statisztikai módszerek szükségesek a maximális keretméret, a szegmenshossz és az átviteli sebesség, független paraméter-kombinációknak a QUIC mechanizmusokra való hatását illetően.

## KÖSZÖNETNYILVÁNÍTÁS

Ezt a munkát a Debreceni Egyetem QoS-HPC-IoT Laboratórium és a TKP2021-NKTA-34 projekt támogatta. A TKP2021-NKTA-34 számú projekt a Kulturális és Innovációs Minisztérium Nemzeti Kutatási Fejlesztési és Innovációs Alapból nyújtott támogatásával, a TKP2021-NKTA pályázati program finanszírozásában valósult meg.

## IRODALMI HIVATKOZÁSOK

- [1] J. Iyengar, M. Thomson: QUIC: A UDP-Based Multiplexed and Secure Transport, *Request for Comments*, RFC Editor (2021), pp. 1-151, doi: 10.17487/RFC9000.
- [2] M. B. Abd-el-Malek, S. S. Hanna: Using filter bank property to simplify the calculations of Empirical Mode Decomposition, *Communications in Nonlinear Science and Numerical Simulation* 62 (2018), pp. 429–444, issn: 1007-5704, doi: <https://doi.org/10.1016/j.cnsns.2018.02.035>.
- [3] R. Bazi, T. Benkedjouh, H. Habbouche, S. Rechak, N. Zerhouni: A hybrid CNNBiLSTM approach-based variational mode decomposition for tool wear monitoring, *The International Journal of Advanced Manufacturing Technology* 5 (2022), pp. 3803–3817, doi: 10.1007/s00170-021-08448-7.
- [4] P. Bloomfield: *Fourier analysis of time series: an introduction*, John Wiley & Sons, 2004.
- [5] F. T. AL-Dhief, N. Sabri, N. A. Latiff, M. Abbas, A. Albader, M. A. Mohammed, R. N. AL-Haddad, Y. D. Salman, M. Khanapi, et al.: Performance comparison between TCP and UDP protocols in different simulation scenarios, *International Journal of Engineering & Technology* 7.4.36 (2018), pp. 172–176.
- [6] K. Dragomiretskiy, D. Zosso: Variational Mode Decomposition, *IEEE Transactions on Signal Processing* 62.3 (2014), pp. 531–544, doi: 10.1109/TSP.2013.2288675.
- [7] B. Eckart, X. He, Q. Wu: Performance adaptive UDP for high-speed bulk data transfer over dedicated links, in: *2008 IEEE International Symposium on Parallel and Distributed Processing*, IEEE, 2008, pp. 1–10.
- [8] Fast transport layer protocol: QUIC, Official web site of IETF QUIC Working Group, url: <https://quicwg.org/>.
- [9] P. Flandrin, G. Rilling, P. Goncalves: Empirical mode decomposition as a filter bank, *IEEE Signal Processing Letters* 11.2 (2004), pp. 112–114, doi: 10.1109/LSP.2003.821662.
- [10] Z. Gál, G. Kocsis, T. Tajti, R. Tornai: Performance evaluation of massively parallel and high speed connectionless vs. connection oriented communication sessions, *Advances in Engineering Software* 157-158 (2021), p. 103010, issn: 0965-9978, doi: <https://doi.org/10.1016/j.advengsoft.2021.103010>.
- [11] S. Garg, M. Kappes: An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11 b networks, in: *2003 IEEE Wireless Communications and Networking*, 2003. WCNC 2003. Vol. 3, IEEE, 2003, pp. 1748–1753.
- [12] J. Gilles: Empirical Wavelet Transform, *IEEE Transactions on Signal Processing* 61.16 (2013), pp. 3999–4010, doi: 10.1109/TSP.2013.2265222.

- [13] D. Griffin, J. Lim: Signal estimation from modified short-time Fourier transform, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.2 (1984), pp. 236–243, doi: 10.1109/TASSP.1984.1164317.
- [14] Y. Gu, R. L. Grossman: UDT: UDP-based data transfer for high-speed wide area networks, *Computer Networks* 51.7 (2007), pp. 1777–1799.
- [15] E. He, J. Leigh, O. Yu, T. A. DeFanti: Reliable blast UDP: Predictable high performance bulk data transfer, in: *Proceedings. IEEE International Conference on Cluster Computing, IEEE*, 2002, pp. 317–324.
- [16] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, H. H. Liu: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences* 454.1971 (1998), pp. 903–995.
- [17] H. K. Rath, A. Karandikar: Performance analysis of TCP and UDP-based applications in a IEEE 802.16 deployed network, in: *2011 The 14th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2011, pp. 1–5.
- [18] G. Rilling, P. Flandrin: One or Two Frequencies? The Empirical Mode Decomposition Answers, *IEEE Transactions on Signal Processing* 56.1 (2008), pp. 85–95, doi: 10.1109/TSP.2007.906771.
- [19] R. C. Sharpley, V. Vatchev: Analysis of the Intrinsic Mode Functions, *Constructive Approximation* 24 (2006), pp. 17–47, doi: 10.1007/s00365-005-0603-z.
- [20] M. Soni, B. S. Rajput: Security and performance evaluations of QUIC protocol, in: *Data Science and Intelligent Applications: Proceedings of ICDSIA 2020*, Springer, 2021, pp. 457–462.
- [21] Y. Yang, J. Deng, D. Kang: An improved empirical mode decomposition by using dyadic masking signals, *Signal, Image and Video Processing* 9.6 (2015), pp. 1259–1263, issn: 1863-1711, doi: 10.1007/s11760-013-0566-7.
- [22] Z. Gal, Gy. Terdik: On the patterns of the nonstationary datagram based fast communication processes, *Annales Mathematicae et Informaticae*, Vol. 58 (2023), pp. 1–15, issn: 1787-6117, doi: 10.33039/ami.2023.08.001.