



# **INTERNET 2022**

The Fourteenth International Conference on Evolving Internet

ISBN: 978-1-61208-975-1

May 22nd –26th, 2022

Venice, Italy

**INTERNET 2022 Editors**

Dirceu Cavendish, Kyushu Institute of Technology, Japan

# INTERNET 2022

## Forward

The Fourteenth International Conference on Evolving Internet (INTERNET 2022) continued a series of events covering the challenges raised by the evolving Internet making use of the progress in different advanced mechanisms and theoretical foundations. The gap analysis aimed at mechanisms and features concerning the Internet itself, as well as special applications for software defined radio networks, wireless networks, sensor networks, or Internet data streaming and mining.

Originally designed in the spirit of interchange between scientists, the Internet reached a status where large-scale technical limitations impose rethinking its fundamentals. This refers to design aspects (flexibility, scalability, etc.), technical aspects (networking, routing, traffic, address limitation, etc.), as well as economics (new business models, cost sharing, ownership, etc.). The evolving Internet poses architectural, design, and deployment challenges in terms of performance prediction, monitoring and control, admission control, extendibility, stability, resilience, delay-tolerance, and interworking with the existing infrastructures or with specialized networks.

We take here the opportunity to warmly thank all the members of the INTERNET 2022 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to INTERNET 2022. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the INTERNET 2022 organizing committee for their help in handling the logistics of this event.

### **INTERNET 2022 Chairs**

#### **INTERNET 2022 Steering Committee**

Renwei (Richard) Li, Future Networks, Futurewei, USA  
Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania  
Terje Jensen, Telenor, Norway  
Przemyslaw (Przemek) Pochec, University of New Brunswick, Canada  
Parimala Thulasiraman, University of Manitoba – Winnipeg, Canada  
Dirceu Cavendish, Kyushu Institute of Technology, Japan

#### **INTERNET 2022 Publicity Chairs**

Mar Parra, Universitat Politècnica de València (UPV), Spain  
Hannah Russell, Universitat Politècnica de València (UPV), Spain

## **INTERNET 2022 Committee**

### **INTERNET 2022 Steering Committee**

Renwei (Richard) Li, Future Networks, Futurewei, USA  
Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania  
Terje Jensen, Telenor, Norway  
Przemyslaw (Przemek) Pochec, University of New Brunswick, Canada  
Parimala Thulasiraman, University of Manitoba – Winnipeg, Canada  
Dirceu Cavendish, Kyushu Institute of Technology, Japan

### **INTERNET 2022 Publicity Chairs**

Mar Parra, Universitat Politècnica de València (UPV), Spain  
Hannah Russell, Universitat Politècnica de València (UPV), Spain

### **INTERNET 2022 Technical Program Committee**

Renwei (Richard) Li, Future Networks, Futurewei, USA  
Abdulraheq Alhammedi, Universiti Teknologi Malaysia (UTM), Malaysia  
Majed Alowaidi, Majmaah University, Saudi Arabia  
Mohammad Alsulami, University of Connecticut, USA  
Mário Antunes, Polytechnic of Leiria & INESC-TEC, Portugal  
Andrés Arcia-Moret, Xilinx, Cambridge, UK  
Marcin Bajer, ABB Corporate Research Center Krakow, Poland  
Michail J. Beliatis, Research Centre for Digital Business Development | Aarhus University, Denmark  
Driss Benhaddou, University of Houston, USA  
Nik Bessis, Edge Hill University, UK  
Maumita Bhattacharya, Charles Sturt University, Australia  
Filippo Bianchini, Studio Legale Bianchini, Perugia, Italy  
Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania  
Fernando Boronat Seguí, Universidad Politècnica De Valencia-Campus De Gandia, Spain  
Christos Bouras, University of Patras, Greece  
Matthew Butler, Bournemouth University, UK  
Alina Buzachis, University of Messina, Italy  
Lianjie Cao, Hewlett Packard Labs, USA  
Dirceu Cavendish, Kyushu Institute of Technology, Japan  
Hao Che, University of Texas at Arlington, USA  
Albert M. K. Cheng, University of Houston, USA  
Hongmei Chi, Florida A&M University, USA  
Andrzej Chydzinski, Silesian University of Technology, Poland  
Franco Cicirelli, ICAR-CNR, Italy  
Victor Cionca, Munster Technical University, Ireland  
Monireh Dabaghchian, Morgan State University, USA  
Noel De Palma, University Grenoble Alpes, France  
Angel P. del Pobil, Jaume I University, Spain

Jun Duan, IBM T. J. Watson Research Center, USA  
Said El Kafhali, Hassan 1st University, Settat, Morocco  
Khalid Elbaamrani, Cadi Ayyad University, Marrakech, Morocco  
Zongming Fei, University of Kentucky, USA  
Steffen Fries, Siemens AG, Germany  
Song Fu, University of North Texas, USA  
Marco Furini, University of Modena and Reggio Emilia, Italy  
Dimitrios Georgakopoulos, Swinburne University of Technology, Australia  
Shadan Golestan, University of Alberta, Canada  
Victor Govindaswamy, Concordia University Chicago, USA  
Shuyang Gu, Texas A & M University-Central Texas, USA  
Wladyslaw Homenda, Warsaw University of Technology, Poland  
Fu-Hau Hsu, National Central University, Taiwan  
Pengfei Hu, VMWare Inc, USA  
Takeshi Ikenaga, Kyushu Institute of Technology, Japan  
Oliver L. Iliev, FON University, Republic of Macedonia  
Khondkar R. Islam, George Mason University, USA  
Terje Jensen, Telenor, Norway  
Sokratis K. Katsikas, Norwegian University of Science and Technology, Norway  
Brian Kelley, University of Texas at San Antonio / 5G Program Management Office - JBSA 5G NextGen, USA  
Ahmed Khaled, Northeastern Illinois University, USA  
Aminollah Khormali, University of Central Florida, USA  
Rasool Kiani, University of Isfahan, Iran  
Lucianna Kiffer, Northeastern University, USA  
Kishori Mohan Konwar, MIT / Broad Institute of MIT and Harvard, USA  
Hovannes Kulhandjian, California State University, Fresno, USA  
Ayush Kumar, ST Engineering, Singapore  
Mikel Larrea, University of the Basque Country UPV/EHU, Spain  
Kevin Lee, Deakin University, Australia  
Kin K. Leung, Imperial College London, UK  
Xin Li, Google, USA  
Zhijing Li, Facebook, USA  
Jinwei Liu, Florida A&M University, USA  
Xiaoqing "Frank" Liu, University of Arkansas, USA  
Pranay Lohia, IBM Research, India  
Luís Miguel Lopes de Oliveira, Institute Polytechnic of Tomar, Portugal  
Olaf Maennel, Tallinn University of Technology, Estonia  
Imad Mahgoub, Florida Atlantic University, USA  
Zoubir Mammeri, IRIT - Paul Sabatier University, France  
Philippe Merle, Inria Lille - Nord Europe, France  
Ivan Mezei, University of Novi Sad, Serbia  
Amr Mokhtar, Intel Corporation, Ireland  
Chan Nam Ngo, University of Trento, Italy  
Jared Onyango Oluoch, University of Toledo, USA  
Carlos Enrique Palau Salvador, Universidad Politécnica de Valencia, Spain  
Fidel Paniagua Diez, Universidad Internacional de La Rioja - UNIR, Spain  
Yanghua Peng, ByteDance Inc., China

Przemyslaw (Przemek) Pochec, University of New Brunswick, Canada  
Mirko Presser, Aarhus University, Denmark  
Marek Reformat, University of Alberta, Canada  
Domenico Rotondi, FINCONS SpA, Italy  
Hooman Samani, University of Plymouth, UK  
Ignacio Sanchez-Navarro, University of the West of Scotland, UK  
Sandeep Singh Sandha, University of California-Los Angeles, USA  
José Santa, Technical University of Cartagena, Spain  
Meghana N. Satpute, University of Texas at Dallas, USA  
Irida Shallari, Mid Sweden University, Sweden  
Mukesh Singhal, University of California, Merced, USA  
Pedro Sousa, University of Minho, Portugal  
Álvaro Suárez Sarmiento, Universidad de Las Palmas de Gran Canaria, Spain  
Diego Suárez Touceda, Universidad Internacional de La Rioja - UNIR, Spain  
Bedir Tekinerdogan, Wageningen University & Research, The Netherlands  
Parimala Thulasiraman, University of Manitoba - Winnipeg, Canada  
Homero Toral Cruz, University of Quintana Roo (UQROO), Mexico  
Mudasser F. Wyne, National University, USA  
Ali Yahyaouy, Faculty of Sciences Dhar El Mahraz, Fez, Morocco  
Ping Yang, State University of New York at Binghamton, USA  
Zhicheng Yang, PingAn Tech - US Research Lab, USA  
Ali Yavari, Swinburne University of Technology, Australia  
Habib Zaidi, Geneva University Hospital, Switzerland  
Huanle Zhang, University of California, Davis, USA  
Yingxuan Zhu, Futurewei Technologies, USA

## Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission to reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

## Table of Contents

Cooperative Caching in Space Information Networks <i>Anders Fongen and Lars Landmark</i>	1
Evaluation of MPTCP with BBR Performance on Wi-Fi/Cellular networks for Video Streaming <i>Masayoshi Kondo, Dirceu Cavendish, Takeshi Ikenaga, and Daiki Nobayashi</i>	6

# Cooperative Caching in Space Information Networks

Anders Fongen  
Norwegian Defence University College (FHS)  
Lillehammer, Norway  
email: anders@fongen.no

Lars Landmark  
Norwegian Defence Research Establishment (FFI)  
Kjeller, Norway  
email: Lars.Landmark@ffi.no

**Abstract**—Members of a Low Earth Orbit (LEO) satellite group need to coordinate their activities in order to improve the quality and timeliness of the services provided to terrestrial clients. As a report from a work in progress, we explore different coordination patterns on a distributed cache and study the hit rate, the training ratio and the space requirements. The query model employs a Scale Free Distribution as this is proven to more accurately model the request patterns to many Internet services.

**Keywords**—LEO satellites; cooperative caching; space information networks; Scale free distribution

## I. INTRODUCTION

A Space Information Network (SIN) is an information system located in space [1]. The concept is an evolution of satellite networks as they are known from the 1960s to present day, where satellites transform from “radio mirrors” with plain wideband transponders, towards networks of interconnected satellites providing connectivity services based on store-and-forwarding of data packets. This evolution represents an *increasing system complexity* in the spacecrafts.

In order for a SIN to provide information and computational services in addition to connectivity services, the theory and methods of *distributed systems* become invaluable tools. The transition from connectivity services to computational services extends the state space of the service sessions, and the distribution and transfer of state, e.g., related to a handover operation, become important fields of study and interesting design problems [2]. Besides, the protection of the new service endpoints occurring in a SIN is essential and requires key and certificate management in the SIN structure [3].

A SIN offering storage services is likely to offer this service as a mutable secondary/slave storage replica, since a data backup needs to exist somewhere. Among many interesting research questions, the problems related to *distributed cache management* are the focus of this paper and will be analyzed and presented in detail.

The performance of *Discovery Services* is an important factors in the efficiency of an information system. These services offer the retrieval, caching and distribution of essential information like X.509 Public Key Certificates, Domain Name System (DNS) name/address pairs, Uniform Resource Locators (URLs), link topology information, etc. Optimal performance of discovery services requires a well balanced and tuned cooperative caching system in the SIN in order to support the relying information services efficiently. E.g., a slow DNS service will hamper the performance of an otherwise well tuned Web service.

Which advantages can be achieved through the deployment of a SIN? Two main characteristics of the services distinguish a SIN from ordinary Internet services:

- 1) Global coverage for mobile clients,
- 2) Very low latency.

The round-trip time through a satellite at 500 km altitude can be as low as 3.3 milliseconds. Low latency is also one key property of 5G, which will enable new time sensitive cooperative applications like remote surgery, autonomous vehicles, etc.

An important choice in our SIN studies is to include the earth’s population density into the analysis and resource planning. In particular for lower altitudes, the satellites will spend large fractions of their time over inhabited areas, mixed with shorter intervals of extremely high density. It is likely that the rate of incoming requests will follow a similar pattern. An appealing idea is to allow idle satellites to offload busy ones, since neighbouring satellites in the network can communicate through high speed inter-satellite links (even optical links).

For the remainder of this paper, the organization is as follows: Section II will present related research on this topic; a discussion on the design of a satellite constellation will follow in Section III. The design of satellite clusters for task distribution will be presented in Section IV, followed by a detailed discussion of the Scale Free Distribution principles in Section V. The experiment series, first based on an isolated cache and next in a satellite constellation, are presented in Sections VI and VII. Finally, some conclusive remarks are given in Section VIII.

## II. RELATED RESEARCH

The term *Space Information Network* has been used to describe networks of satellites and high altitude aircrafts (drones, balloons) with different service levels. Existing satellite networks like Iridium and Starlink [4] offer only communication services, the latter on a very large scale and with high bandwidth. A number of authors have proposed “Cloud Computing in Space” through the addition of larger satellites with sufficient energy and computing resources for taking on these tasks [5][6].

The results presented in this position paper will not deal with technical details in the communication technology, but rather view the SIN as a distributed system which borrows its analysis and solutions from the field of distributed computing. The authors are not aware of other efforts to investigate cooperative caching mechanisms specifically for a SIN.

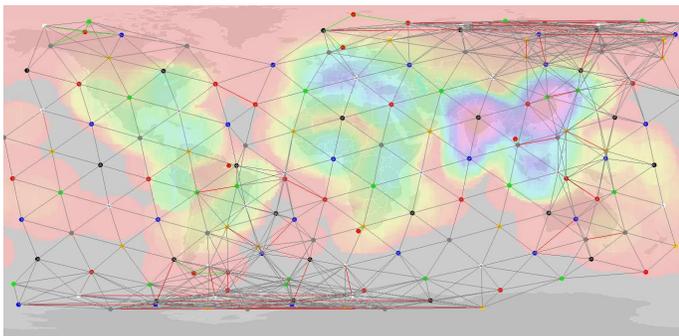


Figure 1. Screenshot from the satellite constellation model

Many retrieval and lookup operations are found to follow a so-called Scale Free Distribution (SFD). Queries for web pages, e-mail addresses, DNS names, etc., have been studied in this regard [7][8]. These results are important for our decision to apply SFD principles in our analyses and experimental design.

### III. A CANDIDATE CONSTELLATION FOR STUDY

Satellite networks servicing civilian mobile clients using handheld equipment tend to operate in a LEO constellation. E.g., the orbit altitude of Iridium satellites is 781 km, which allows for lightweight ground terminals without the need for antenna deployment. The inclination of the orbit can be made so steep that the polar regions are fully covered, or given a lower angle to spend more time over the densely populated latitudes closer to the equator.

The choice of orbit altitude determines the diameter of the *footprint*, e.g., the circular region of the earth with potential connectivity, and the longest possible distance between the satellites which still allows for inter-satellite links and uninterrupted service for ground terminals. Simply stated, a lower orbit altitude reduces the design constraints on the ground terminals and provides higher communication capacity, but increases the cost due to the higher required number of spacecrafts.

For our SIN study, a software model has been made to study these trade-offs and to emulate the coordination activities between the satellites. The model also incorporates population density data which is readily available on the Internet [9].

Figure 1 shows a screenshot from the software model, containing 150 satellites with an orbit inclination of 75 degrees and an altitude of 500 km. The colors on the backdrop indicate the population density inside the footprint of a satellite in that position (contrary to the local density at that exact position). The colorization is considered to be a parameter for the estimation of the request rate received from ground surface clients. Other possible parameters, like local time and Internet penetration of the region, may be taken into regard at a later instance.

The satellites are given a color according to their *role*, which will be explained in Section IV. The lines between them indicate inter-satellite links and links to ground stations.

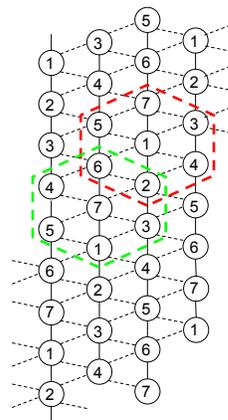


Figure 2. Patterns of relative positions given to satellite roles

Of special interest are links between satellites with the same color, because certain cache optimization techniques will be applied using those links.

In the Iridium system, the 66 satellites are divided into 6 orbits on different longitudes, but with the same inclination. The 6 orbits are separated with 30 degrees and consequently cover one hemisphere in the northbound direction, and the opposite hemisphere in southbound direction. The constellation avoids having the directions interleaved for reasons of handover time and Doppler shift.

Our candidate constellation chooses a similar arrangement, but rather puts the satellites into a spiral arrangement where trailing satellites are shifted eastward to compensate for the earth's rotation. This arrangement makes the row of oncoming satellites to follow the same track when observed from the earth's surface, which will preserve the connection quality across hand-overs.

### IV. DISTRIBUTION OF WORKLOAD ACROSS SATELLITE CLUSTERS

Adjacent neighbors flying in the same directions are keeping company on a permanent basis and may form clusters for distributed processing of requests. The proposed constellation allows for fast and direct links to 6 neighbors (North, South, SE, SW, NE, NW) except for those on the "edge" orbit. Each satellite is given one of seven *roles* numbered 1-7 and they are given relative positions as shown in the pattern on Figure 2. For visualization purposes, the roles are represented using 7 different colors, as shown in Figure 1. The terms *roles* and *colors* are synonyms and will be used interchangeably.

Observe that every satellite is surrounded by the other 6 roles, and that one satellite also serves its role in 6 surrounding clusters. Clusters are not disjoint and every satellite forms the center position of a cluster. Also observe that for satellites at the edge of the constellation, their "missing" neighbor on one side can be found two hops away to the opposite side, through the NE or SW neighbor.

Given this pattern, tasks may be divided into 7 different sub-tasks, and any satellite in the constellation is in a center

position of a cluster which is able to execute it. This satellite may further invoke the resources of its 6 neighbors for the purpose of the task. In the case of a distributed cache, the cache entries may be evenly distributed among these 7 satellites and queries may be delegated to the cache instance which is a candidate for that query value. A hash function modulus 7 computed over the cache entry key is used for this purpose in this experiment.

The motivation for this distributed approach to a caching service is that the total storage capacity increases 7 times, to the cost of a link hop for 6 out of 7 replicas. The performance improvement gained from this design will be evaluated in Sections VI and VII.

## V. A SCALE-FREE DISTRIBUTION OF CACHE REQUESTS

For the evaluation of cache efficiency, a vocabulary of 20,000 words was built and its terms were selected as search terms according to a *Scale Free Distribution* (SFD), also known by the name Zipf's law [10]. If the search term is not found in the cache, it is added to the cache from an authorized source and the operation is counted as a cache miss ( $cm$ ), otherwise a cache hit ( $ch$ ). The performance of the cache is represented by the cache hit fraction of the total number of operations ( $ch/(ch + cm)$ ).

The cache is initially empty, and will generate only cache misses from the beginning. As the cache content is built, its performance gradually improves. When a cache miss results in addition of a term to a full cache, the *Least Recently Used* (LRU) term is removed from the cache to make room for the new term.

The SFD predicts that the observed frequency of a query term is inversely proportional to its *rank*  $r$ . The relative frequency  $f$  of the term  $t$  with rank  $r$  is expressed as

$$f(t_r) = \frac{a}{r} \quad (1)$$

where the value of  $a$  is determined so that

$$\sum_{r=1}^v \frac{a}{r} = 1 \quad (2)$$

and  $v$  indicates the size of the vocabulary in use.

The rationale for preferring SFD over a Uniform Distribution (UD) is that the SFD has been found to provide a good model of different communication and distribution patterns: Email addresses, flight structure between airports, road traffic patterns, sexually transmitted diseases, etc. [11]. The suggestion that a small number of websites have a large portion of the traffic sounds reasonable, and the lookup operations in a DNS cache are shown to be a candidate for an SFD based model [8].

With a vocabulary of 20,000 entries, an SFD will cause one of the 105 highest ranked values to be selected 50% of the time, since the sum of their relative frequencies is 0.5. This indicates that a cache efficiency of 0.5 (meaning 50% hit rate) is theoretically obtainable with only 105 entries in the cache. On the other hand, maximum hit rate with the 1000

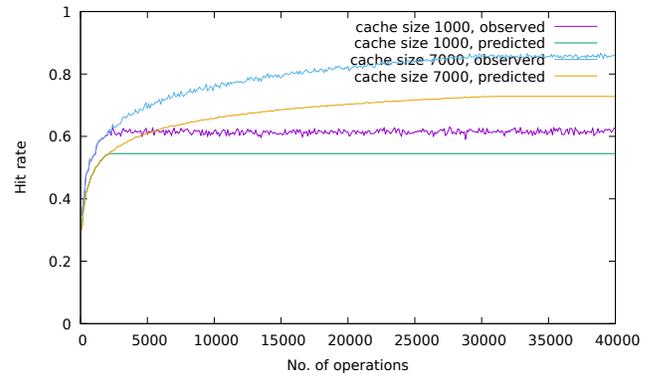


Figure 3. Experimental evaluation of local cache performance

highest ranked terms in the cache is only 0.71. It is therefore expected that from an initially empty cache, the efficiency will rise much faster during the “training phase” than what will be observed using uniform query distribution.

For a cache miss to happen with search term  $t$  at a query operation when  $m$  number of entries are present in the cache, all these entries must contain terms different from  $t$ . Since  $t$  can take any value within the vocabulary, the following sum approximates the cache miss probability  $p_{sfd}(cm)$

$$p_{sfd}(cm) = \sum_{r=1}^v \left( \frac{a}{r} \left( 1 - \frac{a}{r} \right)^m \right) \quad (3)$$

The prediction of cache miss probability with a uniform distribution of search terms can be similarly expressed as a unordered selection operation:

$$p_{ud}(cm) = \frac{\binom{v}{m}}{\binom{v-1}{m}} = \frac{v-m}{v} \quad (4)$$

For further examination, we will use the cache hit probability  $p(ch)$

$$p(ch) = 1 - p(cm) \quad (5)$$

## VI. ISOLATED CACHING EXPERIMENTS

Before looking at the cache performance inside a satellite constellation, the caches are studied in an isolated experimental environment. They expose the predicted behavior, in that they increase their efficiency during “training period” while the cache is filled up, after which only marginal improvements are observed. In Figure 3, the cache hit rate for 1000 and 7000 maximum number of entries is shown with the number of lookup operations along the x axis. The theoretical hit rates are also shown on the figure. They are somewhat lower than the observed numbers. The discrepancy is due to the ratio between vocabulary size and cache size, which Equation 3 assumes to be infinite.

Experimental efforts were made to optimize the performance of the cache through structural modifications, among which modification to the LRU-based replacement algorithm. If replacements increase the *average rank* of the cache entries, a higher hit rate was expected. It was indeed verified that a

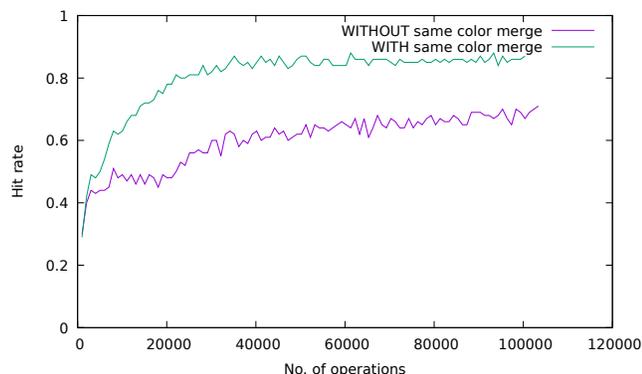


Figure 4. Cache performance inside the satellite constellation

cache with the 105 highest ranked terms obtains a 50% hit rate, but improving the average rank of the 1000 entries in a full cache did not yield significant results over the LRU-based algorithm. The highest possible efficiency of a 1000 element cache is 0.71, as that is the accumulated relative frequency of the 1000 highest ranked terms in the SFD distribution with 20,000 terms. The highest observed hit rate on a 1000 element cache under normal operation using LRU-based replacement algorithm was 0.63; for a 7000 element cache it was 0.86.

This first experiment shows results which are consistent with the theoretical calculations, due to the static data structures in use. In the next section, the more realistic environment of a SIN will be emulated for similar performance evaluations.

## VII. CACHE PERFORMANCE IN A SATELLITE CONSTELLATION

The numbers observed in the experiment presented in Figure 3 serve as a baseline for a distributed and satellite based caching experiment, which will be described in this section.

The environment for a distributed cache residing in orbiting satellites is quite different from the experimental conditions present in the isolated experiment described in Section VI:

- The client does not interrogate the same cluster instance over time, but repeatedly uses different instances “trained” by different clients elsewhere on the planet. The query distribution will change over time and with the geographical region below.
- The cache is distributed over 7 satellites, each of which serves as a member of 7 different cache clusters and is jointly trained by each of them.
- The entire satellite fleet is supposed to hold fully trained caches first after a considerable number of query operations.

The prediction of cache performance under these conditions is likely a non-malleable problem, so a simulation result will be presented here. Seven ground terminals from different parts of the planet surface were configured to send term queries to their nearest satellite, which will contribute to the training of the satellite’s cache and its immediate 6 neighbors. The total average hit rate was calculated and reported as a function of

the total number of queries. Since we now evaluate distributed caches with 7 instances of 1000 entries each, we use the 7000 entries result from the isolated evaluation in Figure 3 as our baseline. The difference in performance will be the effect of the dynamic topology of the satellite infrastructure. The results from this series of experiments are presented as the purple line on Figure 4.

### A. Same color merge

In a third series of experiments, an additional mechanism was added as satellites with the same color/role in the distribution pattern were allowed to merge the content of their caches during periods when they were able to connect. These periods occur when the satellites move across the polar regions, as well as when they meet in opposite directions at the east/west edges of the constellation.

The process of merging caches during an encounter of two same-color satellites will add all entries from one cache into the other, unless it is already there. Since no ranking information is preserved in the cache, they are read from the start, where the higher ranked (and frequently used) entries are to be found. All copied entries will be equally “recently used” since the “recency” of an entry is represented by its position in the linked structure, not by metadata.

The result of this caching technique in the satellite constellation model is shown as the green line in Figure 4. For the 100,000 number of operations shown, the same-color merging process appears to give a significant advantage. Longer experiment runs show that the gap between the two lines narrows down to an insignificant difference after 700,000 operations. Consequently, the merge process merely speeds up the training phase, rather than creating a permanent improvement. The highest hit rate observed is 0.87, which is considered equal to the isolated evaluation with its 0.86 value for best hit rate.

### B. Persistent performance improvement

In a more realistic experiment, the ground stations would choose query terms from different vocabularies, reflecting the diversity in language and culture between the regions of the earth. The vocabulary of query terms from the same region will also change over time, reflecting the changing interests of the population. For a distributed cache to maintain its performance in the presence of constantly changing query term vocabularies, *training speed* becomes an important factor. For this reason, the same-color merge process presented in Section VII-A is more than a ephemeral advantage in the initial phase of operation, but a property expected to yield higher cache performance during the entire operation.

## VIII. CONCLUSION

Despite some simplifying assumption that every client picks query terms from the same distribution and vocabulary, these results show that a cache distributed across a high number of orbiting satellites can achieve the same result as a single instance of the same size. The resulting hit rate is excellent and offers a great reduction in network traffic related to lookup operations. It also shows the successful application of the

LRU-replacement algorithm on scale free distributed term collections. The findings related to cache training speed from Section VII-A are also a welcome contribution to a persistent improvement in the cache performance.

This particular experiment was designed with DNS services in mind, but will equally well predict the results for other SFD collections like e-mail addresses, web pages, X.509 Public Key certificates, collection of shared documents, dictionaries and thesauri, etc.

Future research activities on the SIN model will include the studies of state transitions, handover models, fault tolerance in the presence of failed satellites, routing methods, etc. One important idea in these activities is to consider the population density distribution of the planet in order to even out the workload of the satellites: Computing and communication tasks should be assigned to satellites with less population within its footprint.

#### REFERENCES

- [1] L. Bai, T. de Cola, Q. Yu, and W. Zhang, "Space information networks," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 8–9, 2019.
- [2] A. Fongen, "Application services in space information networks," in *CYBER 2021*. Barcelona, Spain: IARIA, Oct 2021, pp. 113–117.
- [3] —, "Trust management in space information networks," in *SECURWARE 2021*. Athens, Greece: IARIA, Nov 2021, pp. 14–18.
- [4] "Starlink web site," <https://www.starlink.com/>, [Online; accessed 04-Apr-2022].
- [5] S. Briatore, N. Garzaniti, and A. Golkar, "Towards the internet for space: Bringing cloud computing to space systems," in *36th International Communications Satellite Systems Conference (ICSSC 2018)*, 2018, pp. 1–5.
- [6] S. Cao *et al.*, "Space-based cloud-fog computing architecture and its applications," in *2019 IEEE World Congress on Services (SERVICES)*, vol. 2642-939X, 2019, pp. 166–171.
- [7] D. N. Serpanos and W. H. Wolf, "Caching Web objects using Zipf's law," in *Multimedia Storage and Archiving Systems III*, C.-C. J. Kuo, S.-F. Chang, and S. Panchanathan, Eds., vol. 3527, International Society for Optics and Photonics. SPIE, 1998, pp. 320 – 326.
- [8] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS Performance and the Effectiveness of Caching," in *1st ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, November 2001, pp. 153 – 167.
- [9] "Gridded population of the world v4.11," [Online; accessed 04-Apr-2022]. [Online]. Available: <https://sedac.ciesin.columbia.edu/data/collection/gpw-v4/sets/browse>
- [10] G. K. Zipf, *Human behavior and the principle of least effort*. Cambridge, (Mass.): Addison-Wesley, 1949.
- [11] A.-L. Barabasi, *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume Books, April 2003.

# Evaluation of MPTCP with BBR Performance on Wi-Fi/Cellular networks for Video Streaming

Masayoshi Kondo<sup>\*</sup>, Dirceu Cavendish<sup>\*\*</sup>, Daiki Nobayashi<sup>\*\*</sup>, Takeshi Ikenaga<sup>\*\*</sup>

<sup>\*</sup>Graduate School of Engineering, <sup>\*\*</sup>Faculty of Engineering

Kyushu Institute of Technology

Fukuoka, Japan

e-mail: kondo.masayoshi146@mail.kyutech.jp {cavendish@ndrc, nova@ecs, ike@ecs}.kyutech.ac.jp

**Abstract**—Video streaming makes most of Internet traffic nowadays, being transported over Hypertext Transfer Protocol/Transmission Control Protocol (HTTP/TCP). Being the predominant transport protocol, TCP stack performance in transporting video streams has become paramount, specially with regard to multipath transport protocol innovation and multiple client device interfaces currently available. In this paper, we characterize Bottleneck Bandwidth and Round-trip propagation time (BBR) congestion control performance when streaming video over cellular and Wi-Fi access networks, comparing its performance to other available congestion control schemes and path schedulers. We use network performance level, as well as video quality level metrics to characterize multiple path schedulers and resulting network and application layers' interactions.

**Keywords**—Video streaming; TCP congestion control; Multipath TCP; Packet retransmissions; Packet loss.

## I. INTRODUCTION

Data transmission over the Internet relies on transport protocols to control network congestion and avoid uncontrolled data losses. Transmission Control Protocol (TCP) has become the de facto transport protocol of the Internet, supporting reliable data delivery for most applications. Regarding streaming applications, the most dominant type of application over the Internet, stream quality is related to two factors: the amount of data discarded at the client due to excessive transport delay/jitter; data rendering stalls due to lack of timely playout data. It is worth noting that transport delays and data starvation depend heavily on how TCP handles retransmissions during flow and congestion control.

The evolution of portable device hardware, in particular the support of multiple high bandwidth interfaces, has motivated the development of multipath transport protocols, allowing video streaming over multiple IP interfaces and diverse network paths. Multipath video streaming is attractive because it not only increases aggregated device downloading bandwidth capacity, but also improves transport session reliability during transient radio link impairments in handoff situations. An important function in multipath transport is the selection of a path among various active networking paths (sub-flows), which can be done on a packet by packet basis. A path packet scheduler is used for this purpose, and should be designed to prevent head-of-line blocking across various networking paths, potentially with diverse loss and delay characteristics. Head-of-line blocking occurs when data already delivered at the receiver has to wait for additional packets that are blocked at

another sub-flow, potentially causing incomplete or late frames to be discarded at the receiver, as well as stream rendering stalling. As the interplay between path schedulers and TCP variants ultimately define streaming quality, we propose to analyze video performance vis-a-vis popular TCP variants, with attention to BBR [1] and various path schedulers.

The paper is organized as follows. Related work is included in Section II. Section III describes video streaming transport over Transmission Control Protocol, including different TCP variants, one of which BBR. Section IV describes recently proposed alternative path schedulers. Section V characterizes video streaming performance over Wi-Fi and cellular paths via network emulation. We study the performance of a default (shortest delay) path scheduler, as well as alternative schedulers, working with popular TCP variants. Section VI summarizes our studies and addresses directions we are pursuing as follow up to this work.

## II. RELATED WORK

Several multipath transport studies have appeared in the literature, mostly focusing on throughput performance of data transfers over mobile networks (see [2] and related work). More recently, path scheduler research has been recognized as an important driver of multipath transport sessions performance. For instance, [3] has analyzed loss based congestion control TCP variants interactions with minimum Round Trip Time (RTT) default Multipath TCP (MPTCP) path scheduler, and showed how sender/receiver buffer dimensioning impacts throughput performance via inflation of sub-flow RTTs. Little research work, however, has focused on video streaming performance over multiple paths. Motivated by vehicular communication in assisted driving systems, multipath video streaming on ad-hoc network studies have appeared. For instance, [4] introduces an interference aware multipath video streaming scheme in Vehicular Ad-hoc Networks (VANETs). The authors' goal is reliable transport of high quality video streams on vehicle to vehicle communication over multiple paths. Throughput performance is evaluated, taking into account interference within neighbors, as well as shadowing effects onto Signal to Noise ratio, and data delay. They seek to minimize video freezes and dropped frames, via link layer channel interference control, coupled with efficient routing strategies on ad-hoc vehicular networks. In contrast, we focus on video streaming over regular Internet paths, where link layer channel

and route optimization opportunities are limited. Integrated optimization of video and transport layer is proposed by [5], where they introduce a path-and-content-aware path selection approach coupling MPEG Media Transport (MMT) with multipath transport. They estimate path quality conditions of each subflow, and avoid sending I-frames on paths of low transport quality. A similar approach, where different sub-flows are utilized for segregating high priority packets of Augmented Reality/Virtual Reality streams has been introduced by Silva et al. [6]. In contrast, our current work does not couple applications with multipath transport, rather focusing on generic path schedulers and TCP variants to deliver high quality video streaming. About new path schedulers, Ferlin et al. [7] have introduced a path scheduler based on a path head-of-line blocking predictor. They carry out emulation experiments of their proposed scheduler against minimum RTT default scheduler, in transporting bulk data traffic, Web transactions and Constant Bit Rate (CBR) applications. They use goodput, data transport completion time and packet delays as performance evaluation metrics. Still on new schedulers, Kimura et al. [8] have shown throughput performance improvements using schedulers driven by path sending rate and TCP window space, on bulk data transfers. Also, Xue et al. [9] have introduced a path scheduler based on estimation of the amount of data each path is able to transmit. They evaluate the scheduler's throughput performance on simulated network scenarios. In contrast, in our previous works, we have introduced multipath path scheduling generic principles, which can be applied in the design of various path schedulers to specifically improve video stream quality. Using these principles, we have introduced in [10] Multipath TCP path schedulers based on dynamically varying path characteristics, such as congestion window space and estimated path throughput. In addition, in [11], we have also proposed to enhance path schedulers with TCP state information, such as whether a path is in fast retransmit and fast recovery states. Finally, in [12], we have introduced a novel concept of sticky scheduling, where once a path switch is executed, the scheduler stays with the new path until the path bandwidth resources become exhausted. In this work, we evaluate several path schedulers, some of our proposal, in combination with popular BBR TCP [1] over realistic Wi-Fi/Cellular multipath scenarios, focusing on video quality at application layer. We seek to determine whether BBR delivers high performance in combination with path schedulers over multiple network scenarios. BBR performance in MPTCP transport is a novelty. Reference [13] has recently introduced one such study, where BBR is evaluated in combination with a new adaptive packet scheduling scheme (adaptive redundant + predictive). Multiple copies of a packet are injected in paths of low quality, for reliability improvement. The scheme also predicts packet delivery on paths in order to keep in order delivery, mitigating head of line blocking. Their evaluation, however, is limited to throughput and download time performance metrics of files. In contrast, our current study of BBR and schedulers focuses on video streaming quality evaluation.

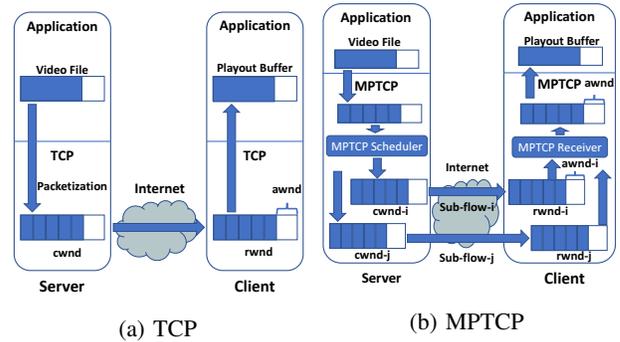


Figure 1. Video Streaming over TCP/MPTCP

### III. VIDEO STREAMING OVER TCP

Video streaming over HTTP/TCP involves a HTTP server storing video content, where video files can be streamed upon HTTP requests over the Internet to video clients. At the transport layer, a TCP variant provides reliable transport of video data over IP packets between the server and client end points. Figure 1 illustrates these video streaming components. As mentioned, HTTP server stores encoded video files. Triggered by a HTTP video request, a TCP sender is instantiated to transmit packetized data to the client machine, connected to the application by a TCP socket. At the TCP transport layer, a congestion window is used at the sender to control the amount of data injected into the network. The size of the congestion window ( $cwnd$ ) is adjusted dynamically, according to the level of congestion experienced through the network path, as well as the space available for data storage ( $awnd$ ) at the TCP client receiver buffer. Congestion window space at the sender is freed only when data packets are acknowledged by the receiver. Lost unacknowledged packets are retransmitted by the TCP layer to ensure reliable data delivery. At the client, in addition to acknowledging arriving packets, the TCP receiver informs the TCP sender about its current available space, so that  $cwnd \leq awnd$  condition is enforced by the sender at all times. At the client application layer, a video player extracts data from a playout buffer, which draws packets delivered by the TCP receiver from its socket buffer. The playout buffer hence serves to smooth out variable network throughput.

#### A. Video Application and TCP Transport Interaction

When tracing data path, at the server side, the HTTP server transfers data into the TCP sender socket according to TCP  $cwnd$  space availability. Hence, the injection rate of video data into the TCP socket is constrained by the congestion condition of the network path, reflected at the  $cwnd$  size, and thus does not follow the video variable encoding rate. On its turn, TCP throughput performance is affected by the RTT of the TCP session over a specific path, since only up to a  $cwnd$  worth of data can be delivered without acknowledgements. Hence, for a given  $cwnd$  size, from the moment a first packet is sent until the first acknowledgement arrives back, the TCP session throughput is capped at  $cwnd/RTT$ . As there are various possible TCP congestion avoidance schemes regulating  $cwnd$ , according to the TCP variant, the size of the congestion window size is computed by a specific algorithm at the time

of packet acknowledgement reception by the TCP source. Regardless of the variant, however, the size of the congestion window computed is capped by the available TCP receiver space  $awnd$ , communicated back from the TCP client, in order to ensure that the receiver buffer never overflows. At the client side, video data is retrieved from the TCP client socket by the video player into a playout buffer, from which data is delivered to the video renderer. Even though client playout buffer may underflow, if TCP receiver window empties out, the playout buffer never overflows, since the player will not pull more data into the playout buffer if no space is available.

### B. Multipath TCP

Multipath TCP is an Internet Engineering Task Force (IETF) extension of TCP transport layer protocol supporting data transport over multiple concurrent legacy TCP sessions [14]. The network multipath transmission of the transport session is hidden from application layer by a legacy TCP socket exposed per application session. Under the hood, at the transport layer, however, MPTCP coordinates concurrent TCP variant sub-flows, each of which in itself unaware of the multipath nature of the application session. In order to accomplish multipath transport, a path scheduler connects the application socket with transport sub-flows, extracting packets from the application facing MPTCP socket, selecting a sub-flow for transmission, and injecting packets into the selected sub-flow. MPTCP transport architecture is depicted in Figure 3(b).

The first and most used path scheduler, called default scheduler, selects the path with shortest Round Trip Time (RTT) among paths with currently available congestion window space for new packets. Other path schedulers have appeared recently. These path schedulers can operate in two different modes: uncoupled, and coupled. In uncoupled mode, each sub-flow congestion window  $cwnd$  is adjusted independently of other sub-flows. On the other hand, in coupled mode, MPTCP scheduler couples the congestion control of the sub-flows, by adjusting the congestion window  $cwnd_k$  of a sub-flow  $k$  according with current state and parameters of all available sub-flows. Although many coupling mechanisms exist, we focus on performance study of Bottleneck Bandwidth and round trip [1], Linked Increase Algorithm (LIA) [15], Opportunistic Linked Increase Algorithm (OLIA) [16], and Balanced Linked Adaptation algorithm (BALIA) [17]. We include also evaluation of uncoupled schedulers recently proposed.

Regardless of path scheduler used, IETF MPTCP protocol supports the advertisement of multiple IP interfaces available between two endpoints via specific TCP option signalling. IP interfaces may be of diverse nature (e.g., Wi-Fi, cellular). A common signalling issue is caused by intermediate IP boxes, such as firewalls, blocking IP options. Paths that cross service providers with such boxes may require Virtual Private Network (VPN) protection so as to preserve IP interface advertising between endpoints. In addition, multipath transport requires MPTCP stack at both endpoints for the establishment and usage of multiple paths.

### C. TCP variants

TCP protocol nowadays have branched into different variants, implementing different congestion window adjustment schemes. TCP protocol variants can be classified into delay and loss based congestion control schemes. Loss based TCP variants use packet loss as primary congestion indication signal, typically performing congestion window regulation as  $cwnd_k = f(cwnd_{k-1})$ , which is ack reception paced. Most  $f$  functions follow an Additive Increase Multiplicative Decrease (AIMD) window adjustment scheme, with various increase and decrease parameters. AIMD strategy relies on a cautious window increase (additive) when no congestion is detected, and fast window decrease (multiplicative) as soon as congestion is detected. TCP NewReno [18] and Cubic [19] are examples of AIMD strategies. In contrast, delay based TCP variants use queue delay information as the congestion indication signal, increasing/decreasing the window if the delay is small/large, respectively. Compound [20] and Capacity and Congestion Probing (CCP) [21] are examples of delay based congestion control variants. Delay based congestion control does not suffer from packet loss undue window reduction due to random, not congestion, packet losses, as experienced in wireless links. Regardless of the congestion control scheme, TCP variants follow a phase framework, with an initial slow start, followed by congestion avoidance, with occasional fast retransmit, and fast recovery phases. BBR congestion control may be considered delay based, since BBR measures the bandwidth and RTT of the bottleneck which a flow goes through [1]. Based on such measurements, BBR adjusts the sending rate to make the best use of the bottleneck bandwidth without dropping its rate during wireless link random losses.

## IV. MPTCP PATH SCHEDULERS

A MPTCP scheduler selects a sub-flow to inject packets into the network on a packet by packet basis. The default strategy is to select the path with shortest average round trip packet delay, hereafter called LRF. If a short and non-congested path exists between the end points, it becomes the preferred path for data transport. Non-congested path is defined as a path for which its congestion window ( $cwnd$ ) has available space among packets yet to be acknowledged by the receiver. Hence, a congested path will have no space for more unacknowledged packets to be injected. Other path schedulers are possible, and the following are evaluated in this paper:

- **Low RTT First (LRF):** In low RTT first, the scheduler first rules out any congested path. Among the non-congested paths, the scheduler selects the path with small smooth RTT ( $sRTT$ ). Smooth RTT is computed as an average RTT of recently transmitted packets on that sub-flow.
- **Largest Packet Credits (LPC):** In largest packet credits scheduler, this scheduler selects the sub-flow with largest available space among the non-congested paths.
- **Largest Estimated Throughput (LET):** In largest estimated throughput scheduler, the scheduler estimates

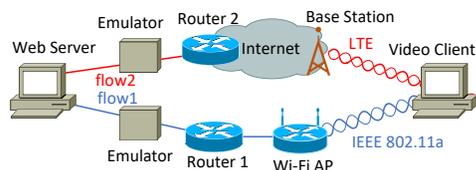


Figure 2. Video Streaming Emulation Network

TABLE I. EXPERIMENTAL NETWORK SETTINGS

Element	Value
Video size	113 MBytes
Video rate	5.24 Mb/s
Playout time	3 mins
Video Codec	H264 MPEG-4 AVC
MPTCP variants	BBR, Cubic, Compound, LIA, OLIA, BALIA
MPTCP schedulers	LRF, LET, LPC, GR-STY, TP-STY, TR-STY

TABLE II. EXPERIMENTAL NETWORK SCENARIOS

Scenario	Emulator	Path properties (RTT)
A1- Baseline (LTE/Wi-Fi)	LTE) delay 0 ms	RTT 80 ms
Scenario A packet loss	Wi-Fi) delay 20 ms	RTT 40 ms
A2- Large delay (LTE/Wi-Fi)	LTE) delay 0 ms	RTT 80 ms
Scenario A packet loss	Wi-Fi) delay 30 ms	RTT 60 ms
B1- Baseline (LTE/Wi-Fi)	LTE) delay 0 ms	RTT 80 ms
Scenario B packet loss	Wi-Fi) delay 20 ms	RTT 40 ms
B2- Large delay (LTE/Wi-Fi)	LTE) delay 0 ms	RTT 80 ms
Scenario B packet loss	Wi-Fi) delay 30 ms	RTT 60 ms

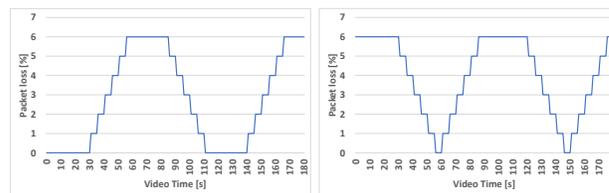
the throughput of each non-congested sub-flow, as  $cwnd/sRTT$ , selecting the one with largest throughput.

- **Greedy Sticky (GR-STY):** As it is the case of default scheduler (LRF), on the onset of a video streaming session, greedy sticky scheduler selects the path with smallest RTT. However, once a new path is selected, the scheduler stays on this new path for as long as there is available congestion window space, until the new path experiences congestion.
- **Throughput Sticky (TP-STY):** Similar to default scheduler (LRF), throughput sticky scheduler selects the path of lowest RTT. However, a new path is selected only if the throughput of the new path is larger than the throughput of the currently selected path.
- **Throughput RTT Sticky (TR-STY):** As with default scheduler (LRF), the path of lowest RTT is first chosen. However, in addition to requiring a larger throughput of a new candidate path as per TP-STY, an extra condition for path switch requires that a new path has smaller RTT than the currently used one.

LPC focuses on scenarios of a large RTT path has plenty of bandwidth, as compared to a shorter RTT path. LET addresses another scenario, in which a short path has plenty of bandwidth. Although the default scheduler may select this path due to its short RTT, if the short RTT has a smaller  $cwnd$ , LET will divert traffic away from this path prior to path congestion, whereas default scheduler will continue to inject traffic through it. Finally, the last three sticky schedulers attempt to reduce the number of path switches during transport session, in order to reduce head of line blocking during the streaming session.

## V. PATH SCHEDULERS OVER WI-FI & CELLULAR PATHS

Figure 2 describes the network testbed used for emulating network paths with Wi-Fi and Cellular (LTE) wireless access links. An HTTP Apache video server is connected to two



(a) Scenario A

(b) Scenario B

Figure 3. Wi-Fi Packet Loss Dynamic Change Scenario

L3 switches, one of which directly connected to an 802.11a router, and the other connected to an LTE base station via an emulator boxes. Since the bandwidth of IEEE 802.11a is sufficiently large for the bit rate of video, we have adopted 802.11a as the wireless LAN interface. In this paper, the emulator boxes are used to vary each path RTT, as well as inject controlled packet losses. The simple topology and isolated traffic allow us to better understand the impact of differential delays, packet loss, TCP variants, and path schedulers on streaming performance.

Network settings and scenarios under study are described in Tables I and II, respectively. Video settings are typical of a video stream, with video playout rate of 5.24 Mb/s, and size short enough to run multiple streaming trials within a short amount of time. Two Wi-Fi packet loss pattern scenarios are used: A and B. Scenario A represents streaming sessions camped at a reliable Wi-Fi network, with occasional packet loss ramp-ups. Scenario B represents an unreliable Wi-Fi network. Emulator boxes are tuned to generate various multiple path network conditions, and have been selected as per Table II to represent commonplace LTE/Wi-Fi streaming situations at home. TCP variants used are: Cubic, Compound, LIA, OLIA and BALIA. Performance measures are:

- **Picture discards:** number of frames discarded by the video decoder.
- **Buffer underflow:** number of buffer underflow events at video client buffer.
- **Sub-flow throughput:** the value of TCP throughput on each sub-flow.

We organize our video streaming experimental results in network scenarios summarized in Table II): A1- A Wi-Fi-Cellular (LTE) baseline with scenarioA packet loss, where Wi-Fi path of low RTT is predominantly used; A2- A Wi-Fi-Cellular scenarioA, where a slightly larger Wi-Fi path delay causes cellular path to be used; B1- A Wi-Fi-Cellular with scenarioB packet loss, where a Wi-Fi link with low delay faces a heavier loss scenario representing user situation at which device is at the end of Wi-Fi range; B2- A Wi-Fi-Cellular scenarioB packet loss, with a Wi-Fi path delay large enough to have cellular path predominantly being used;

### A. A1: Baseline Scenario A

Scenario A packet loss emulates a transition between good to poor Wi-Fi coverage, typical of handoffs between Wi-Fi and cellular when user is leaving home Wi-Fi network.

Figures 4(a) and (b) report on video streaming buffer underflow and picture discard performance when Wi-Fi delay

is 20 ms. Video performance is excellent for BBR, Cubic and Compound TCP variants over all path schedulers. LIA variants perform poorly. Figures 5(a) and (b) report of Cellular and Wi-Fi total packets delivered. We can see that Wi-Fi path is most used for all TCP variants and path schedulers. Moreover, Figures 5(c) and (d) show a larger number of BBR packet retransmissions as compared with other TCP variants across all schedulers in both Wi-Fi and cellular paths.

### B. A2: Large Wi-Fi delay on Scenario A

Figures 6(a) and (b) report on video streaming performance of Wi-Fi - Cellular network scenario with a large 30 ms Wi-Fi path delay. This time, BBR, and Cubic perform well with all schedulers, Compound TCP has a large buffer underflow and picture discard performance using TP-STY scheduler, similar to LIA variants. Throughput performance in Figures 7 (a) and (b) shows that TCP variants of poor video performance under TP-STY prefers Wi-Fi path to LTE path, even under large Wi-Fi path delay and packet loss. In comparison with previous delay case, we see that slowly reactive packet loss TCP variants in combination with throughput driven path selection scheduler delivers bad video streaming experience. Cellular and Wi-Fi retransmissions (c) and (d) are larger for BBR than other variants across most schedulers except TP-STY on Wi-Fi path.

### C. B1: Baseline Scenario B

Scenario B packet loss emulates a transition between poor to good Wi-Fi coverage when user enters the edge of a Wi-Fi network. Figures 8(a) and (b) report on video streaming performance of Wi-Fi - Cellular network scenario with a medium Wi-Fi path delay. We notice a wide variety of performances vis a vis path scheduler/TCP variant combinations. Impressive is the consistent good performance of BBR TCP variant, even across all schedulers. Throughput performance in Figures 9 (a) and (b) reveals a preference of Wi-Fi path for all TCP variants, consistent with Scenario A results presented earlier. Notice, however, that TCP variants that utilizes the most of Wi-Fi path deliver worst buffer underflow video performance, perhaps due to heavy packet losses incurred in the Wi-Fi path. Notice also that BBR maintains a better Wi-Fi utilization, striking a balance between LTE and cellular paths across all packet schedulers. Retransmission data charts, omitted for space sake, show BBR with a significantly larger number of retransmissions across all schedulers than the other TCP variants.

### D. B2: Large Wi-Fi delay on Scenario B

Figures 10(a) and (b) report on video streaming performance of Wi-Fi - Cellular network scenario with a large 30 ms Wi-Fi path delay. As in previous packet loss ScenarioA, the only TCP variants able to deliver good performance across all schedulers is BBR and Cubic. Compound TCP again presents a large buffer underflow performance using TP-STY scheduler. LIA variants all deliver large buffer underflow due to their lack of aggressiveness. Throughput performance in Figures 11 show large LTE path utilization, due to Wi-Fi large delay and heavy

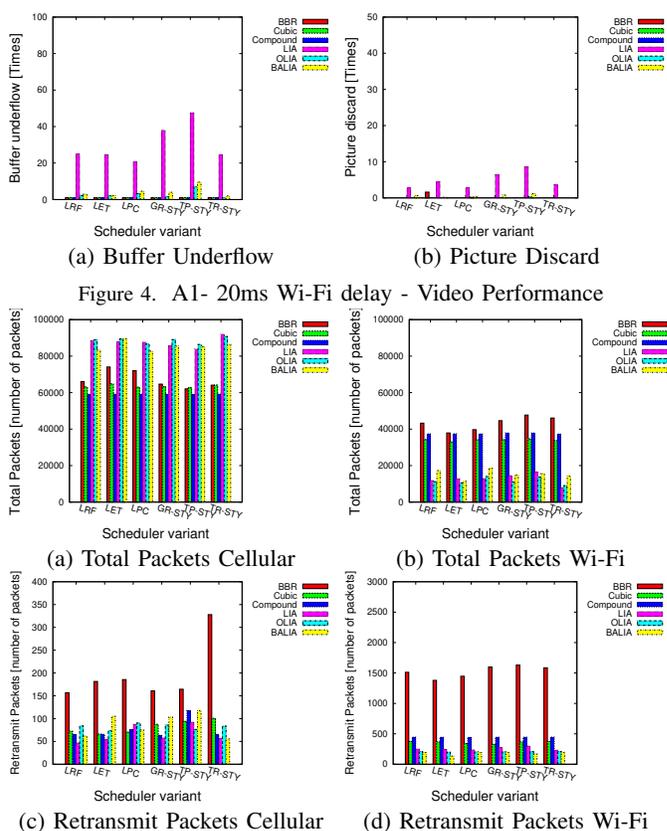


Figure 4. A1- 20ms Wi-Fi delay - Video Performance

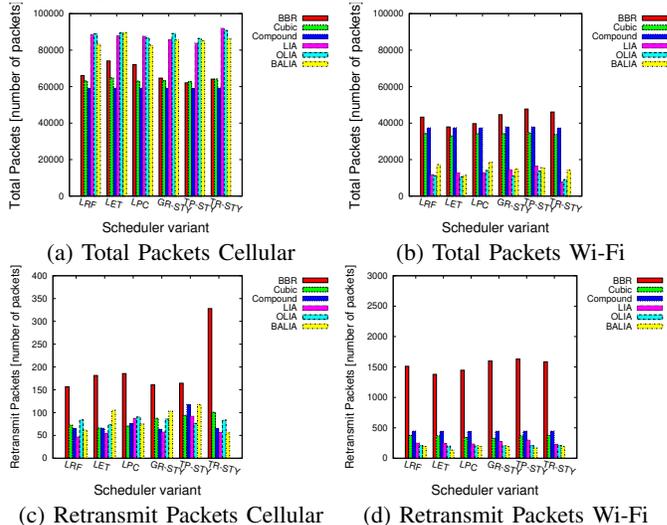


Figure 5. A1- 20ms Wi-Fi delay - Transmission Performance

packet losses across all TCP variants. In comparison with previous delay case, all TCP variants back-off from Wi-Fi path, with BBR still able to use more of the Wi-Fi path across all TCP variants and on most packet scheduler cases. Packet retransmission data, again omitted for space sake, shows a larger number of retransmissions for BBR.

## VI. CONCLUSION

We have provided extensive performance data about path schedulers and TCP variants impact on video streaming performance over multiple paths. We have shown that some combinations of path schedulers with TCP variants negatively impact video streaming performance under certain network scenarios. Noticeable is the impressive performance of BBR across multiple network scenarios and schedulers. Moreover, we have shown video performance degradation for popular LIA and OLIA TCP variants, which can be traced to their congestion adjustment coupling slowing down their packet loss recovery. We are currently investigating the reasons for BBR consistently good streaming performance, despite its large number of retransmissions.

### ACKNOWLEDGMENTS

Work supported by JSPS KAKENHI Grant #21H03430, and National Institute of Information and Communication Technology (NICT).

### REFERENCES

- [1] N. Cardwell, Y. Cheng, I. Swett, and V. Jacobson, "BBR Congestion Control," *IETF draft-cardwell-icrg-bbr-congestion-control-01*, November 2021.

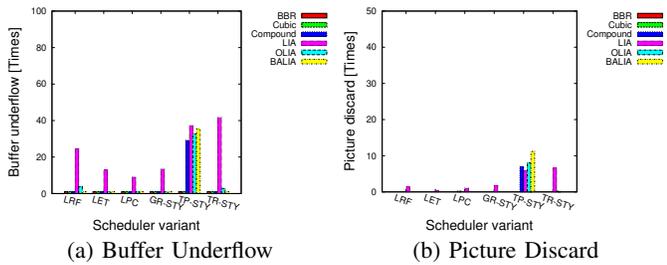


Figure 6. A2- 30ms Wi-Fi delay - Video Performance

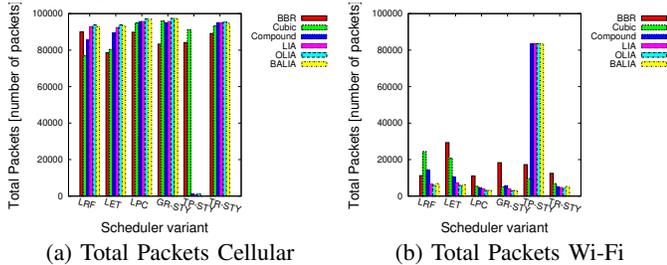


Figure 7. A2- 30ms Wi-Fi delay - Transmission Performance

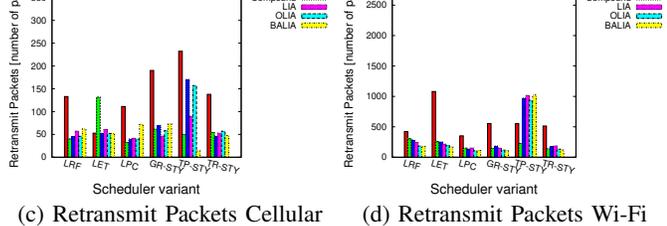


Figure 8. B1- 20ms Wi-Fi delay - Video Performance

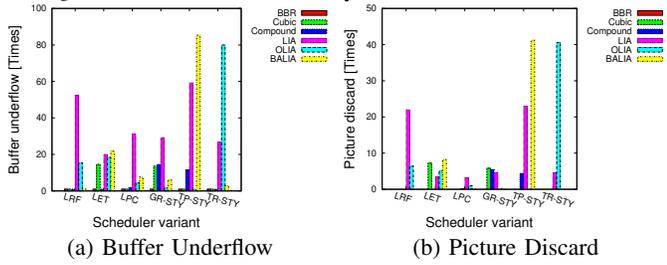


Figure 9. B1- 20ms Wi-Fi delay - Transmission Performance

[2] M. R. Palash et al., "MPWiFi: Synergizing MPTCP Based Simultaneous Multipath Access and WiFi Network Performance," *IEEE Transactions on Mobile Computing*, Vol. 19, No. 1, pp. 142-158, Jan. 2020.

[3] R. Lubben and J. Morgenroth, "An Old Couple: Loss-Based Congestion Control and Minimum RTT Scheduling in MPTCP," *IEEE 44th Conference on Local Computer Networks*, pp.300-307, October 2019.

[4] A. Aliyu et al., "Interference-Aware Multipath Video Streaming in Vehicular Environments," *In IEEE Access Special Section on Towards Service-Centric Internet of Things (IoT): From Modeling to Practice*, Volume 6, pp. 47610-47626, 2018.

[5] S. Afzal et al., "A Novel Scheduling Strategy for MMT-based Multipath Video Streaming," *In Proc. of IEEE Global Communications Conference*

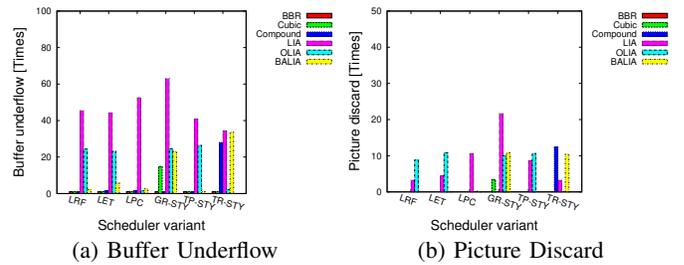


Figure 10. B2- 30ms Wi-Fi delay - Video Performance

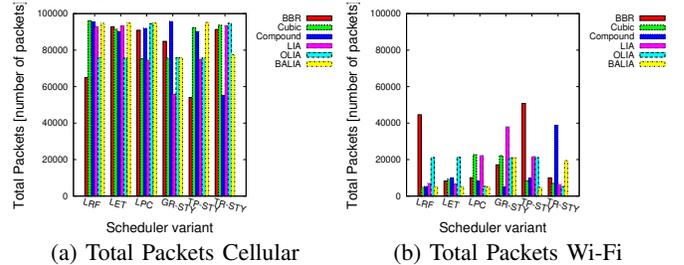


Figure 11. B2- 30ms Wi-Fi delay - Transmission Performance

- *GLOBECOM*, pp. 206-212, 2018.

[6] F. Silva, D. Bogusevski, and G-M. Muntean, "A MPTCP-based RTT-aware Packet Delivery Prioritization Algorithm in AR/VR Scenarios," *In Proc. of IEEE Intern. Wireless Communications & Mobile Computing Conference - IWCMCC 18*, pp. 95-100, June 2018.

[7] S. Ferlin et al., "BLEST: Blocking Estimation-based MPTCP Scheduler for Heterogeneous Networks," *In Proc. of IFIP Networking Conference*, pp. 431-439, 2016.

[8] Kimura et al., "Alternative Scheduling Decisions for Multipath TCP," *IEEE Comm. Letters*, Vol. 21, No. 11, pp. 2412-2415, Nov. 2017.

[9] Xue et al., "DPSAF: Forward Prediction Based Dynamic Packet Scheduling and Adjusting With Feedback for Multipath TCP in Lossy Heterogeneous Networks," *IEEE/ACM Trans. on Vehicular Technology*, Vol. 67, No. 2, pp. 1521-1534, Feb. 2018.

[10] Matsufuji et al., "Multipath TCP Packet Schedulers for Streaming Video," *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, August 2017, pp. 1-6.

[11] Nagayama et al., "TCP State Driven MPTCP Packet Scheduling for Streaming Video," *IARIA 10th International Conference on Evolving Internet*, pp. 9-14, June 2018.

[12] Nagayama et al., "Path Switching Schedulers for MPTCP Streaming Video," *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, August 2019, pp. 1-6.

[13] J. Han et al., "Leveraging Coupled BBR and Adaptive Packet Scheduling to Boost MPTCP," *IEEE Transactions on Wireless Communications*, Vol. 20, No. 11, pp. 7555-7567, November 2021.

[14] A. Ford et al., "Architectural Guidelines for Multipath TCP Development," *IETF RFC 6182*, 2011.

[15] C. Raiciu, M. Handly, and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols," *IETF RFC 6356*, 2011.

[16] R. Khalili, N. Gast, and J-Y Le Boudec, "MPTCP Is Not Pareto-Optimal: Performance Issues and a Possible Solution," *IEEE/ACM Trans. on Networking*, Vol. 21, No. 5, pp. 1651-1665, Aug. 2013.

[17] A. Walid et al., "Balanced Linked Adaptation Congestion Control Algorithm for MPTCP," *IETF draft-walid-mptcp-congestion-control*, 2014.

[18] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," *IETF RFC 2581*, April 1999.

[19] I. Rhee, L. Xu, and S. Ha, "CUBIC for Fast Long-Distance Networks," *Internet Draft*, draft-rhee-tcpm-ctcp-02, August 2008.

[20] M. Sridharan, K. Tan, D. Bansal, and D. Thaler, "Compound TCP: A New Congestion Control for High-Speed and Long Distance Networks," *Internet Draft*, draft-sridharan-tcpm-ctcp-02, November 2008.

[21] D. Cavendish, K. Kumazoe, M. Tsuru, Y. Oie, and M. Gerla, "Capacity and Congestion Probing: TCP Congestion Avoidance via Path Capacity and Storage Estimation," *IEEE Second International Conference on Evolving Internet*, pp. 42-48, September 2010.