## Satellite and Terrestrial Network for 5G

### D4.6

Caching and Multicast – Analysis, Design and Proof of Concepts

<table>
<thead>
<tr>
<th>Topic</th>
<th>H2020-ICT-07-2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Title</td>
<td>Satellite and Terrestrial Network for 5G</td>
</tr>
<tr>
<td>Project Number</td>
<td>761413</td>
</tr>
<tr>
<td>Project Acronym</td>
<td>SaT5G</td>
</tr>
<tr>
<td>Contractual Delivery Date</td>
<td>M12 Interim M27 Final</td>
</tr>
<tr>
<td>Actual Delivery Date</td>
<td>15/12/2019</td>
</tr>
<tr>
<td>Contributing WP</td>
<td>WP4.6</td>
</tr>
<tr>
<td>Project Start Date</td>
<td>01/06/2017</td>
</tr>
<tr>
<td>Project Duration</td>
<td>33 months</td>
</tr>
<tr>
<td>Dissemination Level</td>
<td>Public</td>
</tr>
<tr>
<td>Editor</td>
<td>UoS</td>
</tr>
<tr>
<td>Contributors</td>
<td></td>
</tr>
</tbody>
</table>
Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modifications</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>00.00</td>
<td>10/05/2018</td>
<td>Document creation and ToC</td>
<td>UoS</td>
</tr>
<tr>
<td>01.00</td>
<td>05/07/2019</td>
<td>Initial contributions to sections 3 and 4</td>
<td>UoS</td>
</tr>
<tr>
<td>02.00</td>
<td>23/08/2019</td>
<td>Contribution to section 4</td>
<td>TNO</td>
</tr>
<tr>
<td>03.00</td>
<td>30/09/2019</td>
<td>Contribution to section 2</td>
<td>BPK</td>
</tr>
<tr>
<td>04.00</td>
<td>15/11/2019</td>
<td>Contribution to section 5</td>
<td>ZII</td>
</tr>
<tr>
<td>05.00</td>
<td>11/12/2019</td>
<td>Contribution to section 1 and 6</td>
<td>UoS</td>
</tr>
<tr>
<td>06.00</td>
<td>18/12/2019</td>
<td>Final version after revision</td>
<td>UoS</td>
</tr>
</tbody>
</table>

Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
<th>Contributions include</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chang Ge, Satish Kumar, Ning Wang, Barry Evans</td>
<td>UoS</td>
<td>DASH Live Streaming over Satellite Backhaul; Layered video streaming over SaT5G; Introduction; Conclusion</td>
</tr>
<tr>
<td>Yann Begassat</td>
<td>Broadpeak</td>
<td>Caching &amp; Multicast for Edge Delivery</td>
</tr>
<tr>
<td>Lucia D'Acunto</td>
<td>TNO</td>
<td>Link selection based on MPEG-DASH SAND information</td>
</tr>
<tr>
<td>Leonardo Goratti</td>
<td>ZII</td>
<td>Content Caching and Placement Optimisation at Satellite-backhauled Flights</td>
</tr>
</tbody>
</table>
# Table of Contents

List of Figures .................................................................................................................................. 5  
List of Tables .................................................................................................................................... 6  
List of Acronyms ............................................................................................................................... 7  
Executive Summary ......................................................................................................................... 8  
1 Introduction ...................................................................................................................................... 9  
2 Caching & Multicast for Edge Delivery .......................................................................................... 10  
   2.1 Design overview ....................................................................................................................... 10  
      2.1.1 Offline multicast and caching .......................................................................................... 10  
      2.1.2 Multicast live streaming ................................................................................................. 11  
   2.2 Implementation ....................................................................................................................... 12  
      2.2.1 Offline multicast and caching ........................................................................................ 13  
      2.2.2 Multicast live streaming ................................................................................................. 14  
   2.3 Performance analysis .............................................................................................................. 14  
3 DASH Live Streaming over Satellite Backhaul ............................................................................... 16  
   3.1 Design overview ....................................................................................................................... 17  
   3.2 Implementation ....................................................................................................................... 18  
      3.2.1 Transient Segment Holding and Application-Layer Multicast at MEC Server ............. 18  
      3.2.2 Establishing Transient Segment Holding Policies ......................................................... 19  
      3.2.3 Multicast over Satellite Backhaul ................................................................................... 22  
   3.3 Performance Analysis ............................................................................................................. 22  
      3.3.1 Experiment Setup ............................................................................................................ 22  
      3.3.2 Performance Metrics ....................................................................................................... 23  
      3.3.3 Satellite Backhaul Throughput Performance ................................................................. 24  
      3.3.4 Client-Perceived Throughput Performance ................................................................. 24  
      3.3.5 Client QoE Performance ................................................................................................. 25  
4 MEC-enabled DASH Video Adaptation in Multi-Link Environments ............................................ 29  
   4.1 Link selection based on MPEG-DASH SAND information .................................................. 29  
      4.1.1 Main flow ......................................................................................................................... 30  
      4.1.2 Selecting prefetching candidates ................................................................................. 30  
      4.1.3 Request proxying and segment caching ......................................................................... 31  
      4.1.4 Traffic splitting mechanism ............................................................................................ 31  
      4.1.5 Bandwidth adaptation ...................................................................................................... 32  
      4.1.6 DANE architecture ......................................................................................................... 32  
   4.2 Link selection for SVC Video based on clients buffer level .................................................. 32  
      4.2.1 SVC-HAS Client ............................................................................................................. 33  
      4.2.2 MEC Server .................................................................................................................... 35  
   4.3 MEC based Video-segment Scheduling Network Function ................................................. 38
4.3.1 Request Handler: ............................................................ 39
4.3.2 Adaptation Module: ....................................................... 39
4.3.3 Link Selection Module .................................................... 40
4.3.4 Enforcer ........................................................................ 41
4.3.5 Experiments and Results ................................................. 41

5 Content Caching and Placement Optimisation at Satellite-backhauled Flights ...................... 44
5.1 System model .................................................................... 46
5.2 System analysis ................................................................. 49
  5.2.1 Modelling success probability for the GEO satellite connection .......................... 49
  5.2.2 Modelling success probability for an on-board connection ................................. 52
  5.2.3 Urn models for content caching in an aircraft moving platform ............................. 54
5.3 Results ............................................................................. 56

6 Conclusions ........................................................................ 58

7 References ........................................................................... 59
List of Figures

Figure 3-1 System overview and architecture ................................................................. 17
Figure 3-2 Transient segment holding for multiple clients & application-layer multicast at MEC server .................................................................................................................. 19
Figure 3-3 Satellite backhaul latency over one-hour ....................................................... 20
Figure 3-4 Experiment setup: live 4K streaming via satellite backhaul .......................... 22
Figure 3-5 Satellite backhaul throughput: all streaming scenarios with hold-0 scheme ...... 23
Figure 3-6 Satellite backhaul per-TCP-connection throughput: all streaming and holding scenarios, where H0 refers to hold-0 ................................................................. 25
Figure 3-7 Client-perceived throughput: all streaming and holding scenarios, where H0 refers to hold-0 scheme etc .......................................................... 26
Figure 4-1: DANE in a multilink scenario ........................................................................ 30
Figure 4-2: Selecting prefetching candidates ................................................................... 31
Figure 4-3: Prefetching anticipated segments ................................................................. 32
Figure 4-4: DANE architecture ..................................................................................... 32
Figure 4-5 Content Delivery Framework ........................................................................ 33
Figure 4-6 Performance of SVC-HAS client over Satellite backhaul link .................... 34
Figure 4-7 Client playout buffer State ........................................................................... 35
Figure 4-8 Network Topology Setup for Content Delivery over Emulated Satellite and Terrestrial integrated 5G Network ........................................................... 36
Figure 4-9 Performance of SVC-HAS clients over Satellite link .................................. 37
Figure 4-10 Performance of SVC-HAS clients over satellite and terrestrial integrated 5G network ................................................................. 37
Figure 4-11 Video-segment Scheduling Network Function ............................................ 39
Figure 4-12 Network Topology for Content Delivery over Satellite and Terrestrial integrated 5G Network ................................................................. 42
Figure 4-13 Performance achieved by SVC-HAS clients using VSNF strategy .......... 42
Figure 4-14 Instantaneous Offloading Percentage ......................................................... 43
Figure 5-1: Typical networking configuration for seat screens and personal devices on-board aircrafts ................................................................. 45
Figure 5-2: Future advanced networking on-board an aircraft that integrates with 5G .......... 46
Figure 5-3: End-to-end connectivity scenario in the context of 5G satellite-terrestrial integration ................................................................. 46
Figure 5-4: Two-state model approach to content popularity ....................................... 48
Figure 5-5: General urn model representation in a single urn case ............................... 48
Figure 5-6: Geometric model to compute the bit error probability over the satellite link [29] ................................................................. 49
Figure 5-7: Bit error probability for a Geostationary satellite link in the Ka-band .......... 51
Figure 5-8: Packet success probability computed for the Geostationary satellite link ................................................................. 51
Figure 5-9: Reference scenario to study interference in a heterogeneous access network ......................................................... 52
Figure 5-10: Packet success probability in a heterogeneous access network made of Wi-Fi and LTE unlicensed ................................................................. 54
Figure 5-11: Average number of contents that are cached from the ground versus the popularity index γ for different fractions of the available satellite bandwidth ω ................................................................. 56
Figure 5-12: Average number of contents that are cached close to the passengers versus the popularity index γ and for different fractions of the satellite bandwidth ω when QAM modulation is used at the RTP ................................................................. 57
List of Tables

Table 1 Statistics on key performance metrics’ results ................................................................. 26
Table 2 Encoding bit-rates of the considered videos ................................................................. 36
Table 3 Offloading percentage and avg buffer size with varying \( t_h \) ........................................ 38
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5GIC</td>
<td>5G Innovation Centre</td>
</tr>
<tr>
<td>ABR</td>
<td>Adaptive Bit Rate</td>
</tr>
<tr>
<td>AF</td>
<td>Advance Video Coding</td>
</tr>
<tr>
<td>BDP</td>
<td>Bandwidth Delay Product</td>
</tr>
<tr>
<td>CDN</td>
<td>Content Delivery Network</td>
</tr>
<tr>
<td>CMAF</td>
<td>Common Media Application Format</td>
</tr>
<tr>
<td>DANE</td>
<td>DASH aware Network Element</td>
</tr>
<tr>
<td>DASH</td>
<td>Dynamic Adaptive Streaming Over HTTP</td>
</tr>
<tr>
<td>DN</td>
<td>Data Network</td>
</tr>
<tr>
<td>E2E</td>
<td>End to End</td>
</tr>
<tr>
<td>EGF</td>
<td>Exponential Generating Functions</td>
</tr>
<tr>
<td>EL</td>
<td>Enhancement Layer</td>
</tr>
<tr>
<td>HAS</td>
<td>HTTP Adaptive Streaming</td>
</tr>
<tr>
<td>HLS</td>
<td>HTTP Live Streaming</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>LTE-LAA</td>
<td>LTE Licensed Assisted Access</td>
</tr>
<tr>
<td>LTE-U</td>
<td>LTE unlicensed</td>
</tr>
<tr>
<td>LSM</td>
<td>Link Selection Module</td>
</tr>
<tr>
<td>OTT</td>
<td>Over The Top</td>
</tr>
<tr>
<td>SaT5G</td>
<td>Satellite and Terrestrial Network for 5G</td>
</tr>
<tr>
<td>MEC</td>
<td>Multi-access Edge Computing</td>
</tr>
<tr>
<td>MNO</td>
<td>Mobile Network Operator</td>
</tr>
<tr>
<td>QoE</td>
<td>Quality of Experience</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>SAND</td>
<td>Server and Network Assistant DASH</td>
</tr>
<tr>
<td>SBA</td>
<td>Service Based Architecture</td>
</tr>
<tr>
<td>SNO</td>
<td>Satellite Network Operator</td>
</tr>
<tr>
<td>SVC</td>
<td>Scalable Video Coding</td>
</tr>
<tr>
<td>PED</td>
<td>Personal Device</td>
</tr>
<tr>
<td>POP</td>
<td>Point of Presence</td>
</tr>
<tr>
<td>RTT</td>
<td>Round Trip Time</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>VNF</td>
<td>Virtual Network Function</td>
</tr>
<tr>
<td>VoD</td>
<td>Video on Demand</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>VSNF</td>
<td>Video-segment Scheduling Network Function</td>
</tr>
</tbody>
</table>
Executive Summary

In this work package, we have attempted to investigate the utilization of satellite backhaul links into the 5G system for efficient content caching and multicasting. We have investigated the performance of various use-cases including content caching and multicasting of popular content, video content delivery including live and video on demand application as well as content caching and delivery optimization to moving platforms such as airborne flights. Firstly, we have analysed the performance of Edge local caching and live channel delivery over satellite multicast. For this work, the latency reduction has been observed extensively during the tests and demos, and it was spectacular. Second work deals with Quality of Experience (QoE) assured live video streaming via satellite backhaul in the 5G network. For the very first time in the literature, we carried out experiments and systematically evaluated the performance of live 4K video streaming over a 5G core network supported by a live geostationary satellite backhaul, which validates its capability of assuring live streaming users’ QoE under challenging satellite network scenarios. This work is able to deliver a seamless quality of viewing experience to the end-user by choosing an appropriate holding scheme. The objective of the third work is to develop a video streaming framework over satellite and terrestrial integrated 5G content delivery infrastructure. The mobile edge network is connected to the central 5G core through the terrestrial backhaul network as well as via the satellite backhaul link. In this work, we have proposed content delivery strategies for both Advanced Video Coding (AVC) and Scalable Video Coding (SVC) video. The proposed approaches are able to deliver enhanced quality of video viewing experience to the end-user by optimally utilising both the available backhaul links. Results demonstrated that the proposed strategies are able to deliver stutter-free video with upgraded quality by efficiently utilized both the backhaul links. In the fourth work, we presented a mathematically tractable approach to analyse the caching strategy problem for the case of an aircraft moving platform in which the communication system leverages on 5G technology. We first model the link quality of the satellite connection when a certain fraction of the total bandwidth is made available. Afterward, we modelled the quality of a test radio link that connects a passenger’s personal device to a radio transmission point. Furthermore, we have divided the overall caching strategy problem into two different problems. The first problem that revolves around the amount of content to cache on-board considering the whole availability of contents stored in a ground infrastructure with unlimited resources. The second problem that deals with the amount of content to cache on-board closer to the passengers. Results confirm that the more content can be cached, the better it is since resources can be dedicated to our services, and the caching of content closer to the passengers is mainly driven by the contents’ popularity.
1 Introduction

Satellite networks could be a promising candidate for offloading the terrestrial 5G network traffic due to its high vantage point, high bandwidth and large-scale multi-casting support. However, the efficient utilization of satellite capability in a 5G network for content delivery is still under discussion. Nevertheless, researchers are addressing how to efficiently incorporate the satellite link into the 5G network so that the ever-increasing bandwidth demands of multimedia content can be fulfilled. The objective of this work package is to investigate the effective utilization and integration of satellite backhaul link into the 5G system for optimal content caching and delivery. The use-cases considered in this deliverable are as follows:

- In the first work, two main topics have been investigated for an efficient service delivery over the satellite: Edge local caching and live channel delivery over satellite multicast. Firstly, we have investigated the framework for offline multicasting and caching by utilising a generic architecture of satellite multicast capabilities to cache popular (or expected to be) assets to the edge. Then, we investigated how to optimize the delivery of OTT video live channels. The detailed description of this work is presented in section 2.

- The second work (presented in section 3) addresses the scenarios above by utilizing satellite communications as backhaul in a 5G network to support 4K video streaming applications with QoE assurance. In this work, we focus on an HTTP-based live streaming scenario, where video content is generated on-the-fly at a content origin server and delivered to geographically distributed end-users through a 5G network with satellite backhaul. Specifically, we present a 5G SBA-based framework that provides QoE assurance in a context-aware manner. We envisage that stakeholders are involved in this scenario, i.e., 5G mobile network operator (MNO), video content provider (CP), and satellite network operator (SNO). In the proposed framework, the 5G MNO virtualizes its computing and storage resources and leases them to CPs, where the latter can deploy their own virtualized network functions (VNFs) in multi-access edge computing (MEC) servers [2] [3]. Meanwhile, the SNO leases its satellite channel bandwidth resource to the 5G MNO, so that the latter uses it as a backhaul link in addition to the standard terrestrial backhaul.

- The objective of the third work (presented in section 4) is to develop a video streaming framework over satellite and terrestrial integrated 5G content delivery infrastructure. The mobile edge network is connected to the central 5G core through the terrestrial backhaul network as well as via the satellite backhaul link. In this work, we have proposed content delivery strategies for both Advanced Video Coding (AVC) and Scalable Video Coding (SVC). The proposed methodologies are able to deliver enhanced quality of video viewing experience to the end-user by optimally utilising both the available backhaul links.

Firstly, we have investigated the content delivery framework for AVC video in which we propose an approach where a caching DANE can be highly effective when deployed at the edge of a network that has a multilink connection to the 5G core network. In this approach, the DASH client informs the DANE which segments it is likely to request in the near future. Based on this information, the DANE pre-fetches the video segments using one, or both, of the available links. This allows the DANE to download video segments in parallel (e.g., to increase the overall streaming bitrate), or download video segments over a preferred network link (e.g., to perform network link load balancing or to reduce costs).

Then, we have investigated the content delivery architecture for SVC video which is able to efficiently offload traffic from the terrestrial link to the satellite backhaul link in the 5G system while delivering seamless video viewing experience to all clients. The proposed strategy schedule segment/layer download over both the available backhaul links based on the clients’ buffer status.

- The work presented in section 5 demonstrates a caching strategy to load content on-board aircraft as an application of Multi-access Edge Computing (MEC) use case that 5G features can unlock. In the context of an aircraft, 5G features also refer to the possibility to create a virtualised infrastructure that is open to different service providers or to airlines to instantiate new services more flexibly compared to nowadays. Content has to be made available to the passengers’ PEDs in a different way compared to nowadays in which passengers mainly use their seat screens on a long-haul flight.
2 Caching & Multicast for Edge Delivery

Currently, video content assets (live and video-on-demand – VoD) are served to mobile devices on-the-go from centralised CDNs, usually located on Points of Presence (POPs) owned and controlled by the network operator. It is always a few given POPs that are elected to stream all the content to mobile users. The concept of distributed CDN, where most popular video contents are cached in the edge and streamed from a location closer to end-users, has so far not been used for video content delivery to mobile devices.

There are multiple reasons for that, like:

- the challenge to find physical locations and the cost to deploy and maintain this edge infrastructure
- the overcapacity of aggregation and backbone networks
- the lack of latency-sensitive video applications
- the concentration of the congestion issues at the radio cell level
- the need to distribute mobile core user plane functions at the edge (S/P-GW in 4G, UPF in 5G)

As the video-over-cellular traffic is increasing every year, very soon further boosted by 5G, composed of bandwidth-intensive and latency-sensitive immersive video applications, the central CDNs will not be sufficient anymore.

Leveraging new hosting locations such as base stations or transmission aggregation point would lead to a finer granularity of POPs and the possibility of streaming content from a location closer to end-users. These POPs need to be provisioned with the stream content, with a certain level of elasticity to cache in each location only the most popular ones.

Two main topics have been investigated for an efficient service delivery over the satellite: Edge local caching and live channel delivery over satellite multicast.

2.1 Design overview

2.1.1 Offline multicast and caching

We focused on the delivery of video content asset. The solution envisioned could be easily extended to other types of assets (Webpages, VNF software update repository…). We proposed a generic architecture reusing satellite multicast capabilities to cache popular (or expected to be) assets to the edge (also detailed in deliverable D3.2):
Figure 2-1 Offline caching

Within the Core Network, a dedicated AF for caching is deployed. The AF is in charge of steering traffic to the Edge Network so that it is served locally from the cache server (located in Local DN). This AF is also in charge of orchestrating the caching by multicasting to the edge the popular assets in a carousel. The Local cache joins this multicast and stores locally the assets chunks.

The asset popularity is computed on the fly by an analytics server. For more dynamicity, another solution based on prefetching of segments has also been studied.

Figure 2-2 Dynamic prefetching

The traffic is rerouted to the UPF (RAN), which is connected to a local cache (symbolized here by MEC). This local cache serves the UE directly from its cache and fetches the next segments from the content source before they are even requested by the UE.

2.1.2 Multicast live streaming
Secondly, we investigated how to optimize the delivery of OTT video live channels. As a basis, we reused the work done for the Multicast ABR (mABR). The basic concept behind mABR, is to send in multicast the live traffic over the backbone transmission network (the satellite backhaul in our case) and then convert it back to unicast at the edge of the network. The architecture is very similar to what has been presented for Caching.

Figure 2-3 Live channel delivery

Compared to the generic architecture presented in SaT5G 2018 deliverable D3.1 (Integrated SaT5G General Network Architecture), the following new elements are added:

- A dedicated AF in charge of redirecting the requests to the UPF connected to Function Y;
- A Function X managed which converts live from unicast to multicast;
- A Local DN acting as function Y (in mABR terminology), this function has joined the multicast to receive the live. The Function Y receives the multicast and sends it on as unicast to the UE.

When a UE requests a session on a popular live, this UE is redirected to a decentralized UPF connected to a Local DN. This Local DN receives the popular lives in Multicast from the 5GC and converts them back in Unicast for the UE. The satellite link is used for its inherent Multicast capabilities and efficiencies, and its ability to reach any point across the globe.

2.2 Implementation

SaT5G has helped Broadpeak to design workflows that allow to provision live and VOD content in ABR formats to local edge streamers and to handle the hand-over between several sources of streaming, in the context of an integrated 5G-satellite communication network. The developments of the transcaster (unicast to multicast encapsulator) and nanoCDN agent (multicast to unicast des-encapsulator) has been tested in a satellite context through a prototype implementation.

The implementation has in practice impacted several Broadpeak products:

- The BkE200 unicast to multicast transcaster that pushes the content to a modulator fit for satellite delivery
- The nanoCDN agent (multicast to unicast) that is embedded in a reception server
- The BkS400, the virtualized streaming server that embeds the nanoCDN agent and streams the cached content
- The BkM100, the mediator that configures and monitors the video delivery system, defines the content popularity and its spreading
- The BkA100 analytics server that provides information about content consumption
2.2.1 Offline multicast and caching

As mentioned before, three new elements were added to the generic architecture presented in deliverable D3.1 for the integration of satellite links in the 5G network:

- A cache equipment within the Edge Network titled “Local DN MEC”. This cache is linked to a local UPF which may be elected by the SMF based on traffic steering policies.
- A link between CDN Server and Local DN. This link may be unicast or multicast and is used to populate the local DN with popular assets.
- A new Caching Application Function in the 5G Core: decides on traffic steering policies and provides them to the SMF via PCF.

We will focus here on the scenario where the MNO is in charge of orchestrating the caching of assets within the Local DN (MNO Operated caching). The Caching Application Function holds all the intelligence: it decides on which media content should be pushed, and how the traffic is steered in the 5G Network. An API is shared with the CDN platform; this API may be used by the CDN to indicate the assets they’d like to push.

The “AF” has a leg in both Control Plane and User Plane, it holds the following functions:

- Service Router
- Caching Management
- Multicast Controller

The Service Router uses the information from the Control Plane (Subscriber info, Location, Available Bandwidth, etc.) in order to steer the traffic to the right UPF (through PCF and SMF). The Caching Management function elects and retrieves from the CDN the assets to be pushed to the local cache. This election is based on CDN inputs (through shared API) along with analytics information. The multicast controller based on information from the Caching Management sends the asset to the local cache through a Carousel based multicast.

The detailed caching workflow diagram is presented in Figure 2-4 below.

![Figure 2-4 MNO Operated caching: caching workflow](image)

Two main processes take place, described step by step below:

**Multicast carousel management**

1. The multicast feed which will be joined by the local caches. This multicast is carousel based and contains the assets;
2. Local cache retrieves the assets from the carousel;
3. Local cache stores the missing assets within its cache server.

**Multicast carousel configuration**

1. The CDN platform calls the API provided by the MNO and gives a list of asset sources to be cached. This list contains only the source URL for the assets;
2. NEF stores this information within the UDR;
3. Periodically the Caching Management function retrieves the list of asset sources that the CDN would like to be cached from the UDR;
4. Caching Management retrieves the list of currently popular assets from the analytics. How the analytics determines what is popular is out of scope (can be based on session counting, user location…);
5. Caching management merges both information (Step 3 and 4) and regenerates the carousel to be sent in the multicast. If there are missing assets in its local storage then it retrieves them from the CDN source and stores them locally;
6. Carousel is provided to the Multicast controller which in turn modifies the multicast with the new carousel.

### 2.2.2 Multicast live streaming

As we said in 2.1.2, the Multicast live streaming workflows are equivalent to the ones of offline caching.

**Figure 2-5 mABR SaT5G Workflow**

Workflows can be briefly described as follow:

**Live multicast management**

1. The Multicast Controller pulls from the origin the live in unicast;
2. Multicast Controller generates a multicast flow which is joined by the Local DN. Local DN serves the live in unicast.

**Live multicast configuration**

1. Periodically, the Live Management checks from its analytics server the list of popular live content;
2. Then it checks internally if some live content is expected to be popular and should therefore be multicast;
3. Based on this information the Live Management configures the Multicast Controller to create a new multicast for those live content.

### 2.3 Performance analysis
Caching and multicast streaming performance shall be measured against three KPIs:
- network bandwidth savings
- latency reduction
- video quality

While the technical setup that was used in our project could not allow us to run a thorough performance analysis, we have been able to make some interesting observations on latency and we can remind some important principles with regards to performance as well.

1. The multicast transmission benefit is intrinsic and immediate, and its performance in terms of network bandwidth savings fundamentally relies on the channel popularity statistical distribution. We typically observe that the 20 most popular channels command around 80% of the viewership. Therefore, offloading these 20 channels on the satellite link allows to divide the required capacity on the landline broadband network by 5.

2. The latency reduction has been observed extensively during the tests and demos, and it was significant. The deterministic nature of multicast traffic allows to significantly reduce the terminal player’s buffer size, so we could regularly see latencies between 5 and 10 seconds while “normal” OTT video systems deliver latency figures between 20 and 60 seconds. Combined with low-latency streaming protocols (e.g. CMAF/CTE, HLS), multicast ABR can even deliver a latency of 1-2 seconds (between transcoder output and display on a screen), which is similar to what we observe in broadcast (Direct-to-Home – DTH – satellite and cable for instance).

3. Multicast ABR has a very positive impact on video quality thanks to its capability to deliver the highest video profile 99% of the time, where this statistic spans between 50% and 90% in standard OTT systems. In the latter, the available throughput and thus the served video profiles depend on the number of concurrent users, which is by definition not the case with multicast ABR.
3 DASH Live Streaming over Satellite Backhaul

In this Work, under the EU 5GPPP Phase-2 SaT5G (Satellite and Terrestrial network for 5G) project [1], we address the scenarios above by utilizing satellite communications as backhaul in a 5G network to support 4K video streaming applications with QoE assurance. We focus on HTTP-based live streaming scenario, where video content are generated on-the-fly at a content origin server and delivered to geographically distributed end-users through a 5G network with satellite backhaul. Specifically, we present a 5G SBA-based framework that provides QoE assurance in a context-aware manner. We envisage that stakeholders are involved in this scenario, i.e., 5G mobile network operator (MNO), video content provider (CP) and satellite network operator (SNO). In the proposed framework, the 5G MNO virtualizes its computing and storage resources and leases them to CPs, where the latter can deploy their own virtualized network functions (VNFs) in multi-access edge computing (MEC) servers [2] [3]. Meanwhile, the SNO leases its satellite channel bandwidth resource to the 5G MNO, so that the latter uses it as a backhaul link in addition to the standard terrestrial backhaul. The framework’s operations include the following aspects:

- For each user, its video requests are handled by a local CP-operated VNF that is hosted at the MEC server. Such a MEC server is envisaged to be located within proximity of the user to ensure low access latency. It also breaks the end-to-end (E2E) content delivery path into two segments, i.e., satellite backhaul and 5G Radio Access Network (RAN). This not only significantly reduces initial startup delay, but also offers better content delivery performance [4] [5].
- Each MEC server monitors the active live streaming sessions under its coverage, especially their real-time QoE status such as buffering and video quality. Based on these contexts, it performs necessary content operations such as transient segment holding [5] etc, which provide customized, context-aware QoE assurance on a per-user, per-session basis. Meanwhile, it may perform transport-layer performance enhancement techniques to complement the application-layer QoE assurance above. Furthermore, a MEC server aggregates requests from all sessions that consume the same live stream under its coverage. In other words, regardless of how many sessions are consuming a stream, only one flow needs to be established between that MEC server and the content origin, which effectively realises application-layer multicast in the last hop.
- Each CP monitors its live video streams’ spatial popularity, and dynamically adjusts the method that a live stream is delivered from its origin server to the MEC servers. For example, if a live stream is detected to be popular among multiple distributed MEC servers, the CP may decide that it is more efficient for the content origin to use a multicast-based protocol such as FLUTE (File Delivery over Unidirectional Transport [6]) to disseminate the live stream files to multiple MEC servers, which is inherently supported over the satellite backhaul. If the stream’s popularity drops, the CP may switch back to unicast-based delivery from origin to MEC servers. This realises context-aware backhaul multicast and ensures that backhaul content traffic consumes bandwidth efficiently.

The key contributions of this work are as follows:

- This is the first system developed in the literature that utilizes both SBA-based 5G core network and satellite backhaul to support 4K HTTP-based live streaming applications with QoE assurance. Specifically, it leverages both the context awareness and flexibility that are enabled by 5G SBA architecture, as well as the multicast capability of satellite backhaul. It also utilizes virtualization technology to enable CPs to deploy their own VNFs in MEC servers at 5G mobile edge, which not only performs content operations such as transient segment holding, but also realises last-hop multicast at application layer. Overall, the proposed system assures live users’ QoE while maintaining the video quality at or above 4K; it also ensures that video content are always delivered through the backhaul in the most efficient manner.
- This is the first time that a 5G core network and a real satellite communications link have been implemented and integrated as a holistic system, where the latter serves as the backhaul of the 5G network. The establishment of such a system means that it is possible to test the performance of MEC servers with content operations (such as transient segment holding) in terms of content delivery and QoE assurance through a real satellite backhaul.
Based on the implemented system above, we have comprehensively evaluated the performance of our QoE assurance scheme in a wide variety of scenarios over-the-air using a satellite backhaul integrated with a 5G core network. The experiment outcome validates that even through satellite backhaul, the proposed scheme is able to guarantee a stalling-free live streaming experience while maintaining the video quality at 4K.

3.1 Design overview

In this section, we provide a high-level overview of the proposed system architecture, which is shown in Figure 3-1. There are three stakeholders in the system, i.e., 5G MNO, SNO and CP. In summary, a live streaming user’s E2E content delivery path involves an edge User-Plane Function (UPF, which typically runs on a MEC server), satellite backhaul and a core UPF before it reaches the CP’s live content origin that is located in the Data Network (DN). The satellite backhaul involves a satellite terminal, the satellite link and a satellite gateway. The 5G MNO leases satellite channel capacity from the SNO to use it as a backhaul link. Note that although the MEC server is operated by the 5G MNO, its computing and storage resources are virtualized and leased to CPs who deploy their own VNFs to perform content operations such as caching and transient segment holding. As shown in Figure 3-1, a MEC server can host multiple VNFs that are operated by different CPs. Such an operational model has been adopted in the operator and content delivery industries since 2015 [7].

![Figure 3-1 System overview and architecture](image)

From the User Plane perspective, when a user sends an HTTP request in a live streaming session, it is first resolved to the corresponding CP’s VNF within the MEC server. The VNF intercepts the request and acts as a reverse HTTP proxy. Through content operations such as transient segment holding, the MEC server can download video segments from their origin server and cache them locally before they are requested by users. This is achieved through manipulating the live stream’s manifest file content and establishing multiple parallel TCP connections from the MEC server towards the origin server, and more details are described in Section 3.1. If the requested video segment is already available at its local cache, it is served to the user immediately. If not, the VNF forwards the requests to the content origin and retrieves it on the user’s behalf. Meanwhile, besides normal content operations such as caching, it also performs context-aware operations such as transient segment holding while subjecting to the CP’s policy. More details on how such policies are established are described in Section 3.2. The VNF is also responsible for monitoring and reporting application context (e.g., each user’s real-time QoE status) periodically to the Control Plane.

Regarding the Control Plane, it is shown in Figure 3-1 that it adopts a SBA where all its elements communicate via a bus. This is in line with the latest 3GPP 5G system architecture [8]. In this work, we focus on five elements that are most relevant to our scheme, namely Application Function (AF), Policy Control Function (PCF), Session Management Function (SMF), Network Exposure Function (NEF) and Access and Mobility Management Function (AMF). While most Control Plane elements are operated by the MNO, AF is typically operated by third-party stakeholders such as CPs. Their functionalities are as follows:
• PCF is responsible for converting instructions from other Control Plane elements (e.g., SMF and AF) into policies that can be understood by CPs' VNFs.
• SMF is the direct "contact" point between the MEC server (and the VNFs within) and the Control Plane. On one direction, SMF receives policies from PCF and disseminates them to the VNFs where they will be enforced. On the other direction, SMF handles context updates and monitoring feedback from VNFs, and sends them to AF so that it can update its policies as necessary.
• AF has multiple functionalities. First, it provides context information to the MNO on how to identify each CP's traffic flow. Example criteria include destination IP address and port number. Second, it processes the monitored contexts that are reported by the MEC server via SMF, and adjusts content operation (e.g., caching or segment holding etc.) policies where necessary. Updated policies are sent to PCF first, where they are formatted and sent to SMF for dissemination and enforcement.
• NEF acts as a "bridge" between AF and other Control Plane elements. Because AF is operated by third-party stakeholders, the MNO needs to carefully control which aspects of the Control Plane are exposed to AF, and NEF is responsible for managing such exposure policies.
• AMF is mainly responsible for the management aspects of the Control Plane, such as authentication, authorization and user mobility management etc. For example, an AF needs to be first authenticated by AMF before it can interact with other Control Plane elements such as SMF.

Transient segment holding for multiple clients & application-layer multicast at MEC server

3.2 Implementation

3.2.1 Transient Segment Holding and Application-Layer Multicast at MEC Server

As mentioned in Section 3.1, in order to assure live streaming users' QoE at the MEC server, context-aware transient segment holding (first proposed in [5]) is performed on a per-streaming-session-basis. In a nutshell, a live streaming client periodically requests the video stream's manifest file from the live origin, so that it can learn about newly-produced segments as soon as possible. Since all requests for stream manifest and video segment files are handled by the MEC server, if it holds back the availability of some segments from the client, the client would be given the false impression of the live origin's streaming progress and request segments that were produced a small while ago. This creates the opportunity for the MEC server to download those held-back segments from the live origin beforehand using parallel TCP connections, hence making them available locally before requested by the client. Intuitively, while holding more segments will provide better assurance on video segment localization, it also introduces higher live streaming latency at the client side. Therefore, the technical challenge is to minimize the number of held segments, while assuring that all video segments are localized before they are requested by clients. More details on how such optimization is performed over a satellite backhaul are specified in Section 3.2.2.

By default, the above techniques are performed on each individual live streaming session that is requested through the MEC. However, if a live stream is popular, it is expected that multiple clients will be consuming it simultaneously under a single MEC server's coverage. Although their streaming progresses are typically similar, there may be a difference of one or two segments between clients' progresses due to e.g., minor clock drifting on user devices. This is especially the case with shorter segment lengths, e.g., 2s or 4s. In order to handle such a situation, for each live stream, the MEC server always aligns its segment holding progress to the session with the most advanced streaming progress. This is illustrated in Figure 3-2. Note that for clarity, we omit all manifest requests (besides the first one) and all response flows in the figure.

In the beginning, the MEC server performs transient holding of two segments for client 1 that joins a live stream, where it opens three parallel TCP connections to download segments 35 to 37 simultaneously. Afterwards, client 2 joins the same stream, and its progress is one segment behind client 1. In this case, the MEC server directly serves the segments that were already made available locally for client 1, and there is no need for extra actions. Later, session 3 joins the same stream with
its progress two segments ahead of session 1. In this case, the MEC server first serves segment 39 that was already pre-downloaded for client 1. Meanwhile, it downloads segments 40 and 41 to ensure that it always stays two segments ahead of client 3’s progress, which also ensures that all other sessions will also have access to locally available segments for their subsequent requests. Note from Figure 3-2 that the MEC server adjusts its downloading schedule to match client 3’s progress over client 1’s as soon as it detects client 3 is the one with the most advanced progress.

![Figure 3-2 Transient segment holding for multiple clients & application-layer multicast at MEC server](image)

It is worth noting that although not depicted in Figure 3-2, there are three possible scenarios regarding the availability of a video segment at the MEC server when it is requested by a user. It may a) have fully downloaded the segment already, or b) have started downloading it but the transmission is not finished yet; or c) have not started the download yet. Scenario b) is more likely to happen when the MEC server is using multiple parallel TCP connections to download several segments simultaneously, as their transmissions do not necessarily finish in order.

One key observation from Figure 3-2 is that each video segment file is only downloaded once by the MEC server from the live video origin. This is neither affected by the number of clients that consume a live stream, nor by how much their streaming progresses differ. Henceforth, our scheme has effectively realized application-layer multicast at the MEC server. It not only aggregates all clients’ requests for each video stream into a single flow between itself and the live video origin, but also dynamically adjusts its transient segment holding schedule to ensure all clients always have access to locally available video segments. This is especially important for satellite backhaul where bandwidth resources are limited.

### 3.2.2 Establishing Transient Segment Holding Policies

As described above, the main challenge when the MEC server establishes its transient segment holding policies for each session is to determine the optimal (i.e., minimal) number of segments to be held. Such a problem has been formulated in [5], where the objective is formulated as follows:

$$\arg\min_x \frac{s_{\text{seg}}}{l_{\text{bh}, x}} \leq l_{\text{seg}} \quad (x = 1, 2, \ldots)$$  \hspace{1cm} (1)
subject to:

\[ t_{\text{ran}} > b_{\text{seg}} \]  

where \( x \) denotes the number of segments to be held. \( s_{\text{seg}} \) and \( l_{\text{seg}} \) refer to the size (in bytes) and the length (in seconds) of a video segment respectively. \( t_{\text{bh}} \) refers to the backhaul throughput between the MEC server and the live video origin, i.e., the throughput over the satellite link. Objective (1) means that each video segment’s download duration must be shorter than its length, hence ensuring that it is available at the MEC server before it is requested by a client. Note that the overall aggregated backhaul throughput is \( t_{\text{bh}} \cdot x \) because \( x \) parallel TCP connections will be used to download \( x \) segments simultaneously. Constraint (2) assumes that RAN throughput \( t_{\text{ran}} \) is always greater than the video segment’s bitrate \( b_{\text{seg}} \), which ensures that as long as a video segment is available at the MEC server, it can always be delivered to the client over RAN in time. In this work, we assume this constraint always holds true\(^1\).

\[ \text{Figure 3-3 Satellite backhaul latency over one-hour} \]

In objective (1), besides the variable \( x \) which needs to be minimized, the only field that is not directly known is \( t_{\text{bh}} \). In other words, we need to model the throughput that the MEC server gets when downloading a video segment from the live origin via the satellite link. In this work, we follow the modeling approach in [5] that is based on the recently-proposed BBR (Bottleneck-Bandwidth and Round-trip Latency) as the TCP congestion control mechanism [9] [10], which is due to its superior performance over links with large bandwidth-delay product (BDP) as well as relatively high packet error rates such as satellite links. What is different from [5] is that while fixed backhaul network exhibits a very stable BDP, satellite backhaul has distinct BDP characteristics. We have performed measurements of latency and jitter over a real satellite link for one hour, and the round-trip time (RTT) results are plotted in Figure 3-3. It is observed that while not as stable as fixed backhaul links, the satellite latency is still relatively stable as 99.7-percentile of the results fall within 540ms to 580ms. Statistically, the latency measurement has a mean of 560.03ms with a standard deviation of 13.02ms. The bandwidth on a satellite link is relatively stable as well. Therefore, we make the following assumptions:

- Each video segment’s size is relatively small, and can usually be fully delivered within 10-20 RTTs based on our measurements above.

\(^1\) It is explained comprehensively in [5] what the MEC server can do if it does not hold.
The backhaul RTT is relatively stable with low jitter, hence the overall BDP does not fluctuate significantly.

Packet loss over the satellite backhaul does not affect BBR's performance, since BBR is a rate-based (instead of loss-based) congestion control mechanism and adjusts its sending rates via BDP measurement.

Based on these assumptions, we model $t_{bh}$ over the satellite backhaul link as follows. First, for each video segment download session, the number of bits that is transmitted during the TCP BBR's startup phase is modeled as $s_{\text{startup}}$:

$$s_{\text{startup}} \approx \frac{\text{mss}(1-\frac{2}{\ln(2)})^{d_{\text{startup}}}}{1-\frac{2}{\ln(2)}}$$  \hspace{1cm} (3)

where mss refers to TCP's maximum segment size (e.g., 1460 bytes), and $d_{\text{startup}}$ denotes the time duration that is spent in BBR's startup phase. Note that by default, BBR begins by increasing its sending rate with a factor of $2/\ln(2)$ and stops when no more additional bandwidth is found after three RTTs. Therefore, we have

$$d_{\text{startup}} = (\log_2\text{BDP} + 3) \cdot \text{RTT}$$  \hspace{1cm} (4)

where BDP is estimated by multiplying satellite link bandwidth $bw_{bh}$ by RTT.

Since the satellite backhaul's BDP is generally stable, the total transmission duration of a video segment $d_{\text{seg}}$ is modeled as:

$$d_{\text{seg}} \approx d_{\text{startup}} + \frac{(s_{\text{seg}}-s_{\text{startup}})}{bw_{bh}}$$  \hspace{1cm} (5)

which indicates that the remaining amount of bits that were not transmitted during the startup phase will be downloaded at $bw_{bh}$. The overall backhaul throughput can be hence modeled as:

$$t_{bh} = \frac{s_{\text{seg}}}{d_{\text{seg}}}$$  \hspace{1cm} (6)

Substituting into the objective function (1), we get:

$$\arg\min_x \ x \geq \frac{1}{t_{seg}} (d_{\text{startup}} + \frac{(s_{\text{seg}}-s_{\text{startup}})}{bw_{bh}})$$  \hspace{1cm} (7)

It is worth noting that while bandwidth is typically over-provisioned in fixed backhaul links and is in the order of hundreds to thousands of Mbps, it is much lower in a satellite link. For example, as we will show in Section 4, the TCP throughput that can be maximally achieved over a typical 20MHz Ku-band satellite channel is generally less than 60Mbps. Furthermore, when transmitting over such a bandwidth-limited link, opening multiple parallel TCP connections will lead to degraded per-connection throughput performance and even channel congestion. Therefore, although transient segment holding’s strategy is to use parallel TCP connections to compensate a single connection’s poor performance, cautions must be taken to avoid opening too many parallel TCP connections from over-saturating the satellite channel and causing congestion on the link. In Section 4, we will evaluate the minimum number of parallel TCP connections that are needed under various scenarios, as well as multiple TCP connections’ impacts on throughput performance.
3.2.3 Multicast over Satellite Backhaul

So far, we have described how each MEC server aggregates its downstream requests from clients, and establishes one transmission flow per live stream towards the live origin. Meanwhile, it is expected in a 5G network that MEC servers are deployed in a distributed manner among potentially numerous mobile edge sites. If a live stream is popular among multiple MEC servers, they would be delivered from the live origin to every MEC server via unicast by default, which multiplies the traffic on the satellite backhaul with limited bandwidth resource.

In this work, we propose that if a live stream is being consumed by multiple MEC servers simultaneously, the live stream’s origin should deliver the segment via multicast, instead of unicast, to these MEC servers. The key rationale here is to significantly reduce traffic volume over the satellite backhaul for popular live streams [11]. This is especially beneficial to 5G networks where MEC servers are envisaged to be deployed close to end-users and hence, a significant number of MEC servers will be required. Furthermore, in the case of live streaming, the MEC servers’ requesting progress are likely to be very similar to each other’s, making it an ideal scenario for push-based file deliveries. Existing multicast-based content delivery mechanisms such as [6] and [12] can be used to transmit the files while assuring file integrity, hence eliminating the risk of corrupting video frames.

As mentioned in Section 2, each MEC server regularly monitors and reports popularities of live streams under its coverage to SMF in the control-plane, which then forwards such context updates to the AF that is operated by CPs. If the AF detects that a live stream is being consumed by multiple MEC servers, it will send a policy update to the corresponding MEC servers via PCF and SMF, which contains instructions for them to join the respective live stream’s multicast group at the origin. For example, the AF may tell the MEC servers about which live streams are available through which multicast groups, as well as optional recommendations or policies on which groups they should join. Note that the methodology above is just one example of how to achieve the proposed scheme, and the CP may flexibly determine the specific operating policy on this. For example, the CP may advertise one multicast group per live stream, or it may multicast multiple live streams simultaneously if they are often consumed together.

![Experiment setup: live 4K streaming via satellite backhaul](image)

Figure 3-4 Experiment setup: live 4K streaming via satellite backhaul

3.3 Performance Analysis

In this section, we evaluate the performance of transient segment holding mechanism and its QoE assurance in a variety of scenarios using a real satellite backhaul.

3.3.1 Experiment Setup

The experiment setup is illustrated in Figure 3-4. At the client side (site A), a laptop uses Chrome browser (version 68) with hls.js v0.11.1 as the HLS (HTTP Live Streaming) client. It connects to a local WiFi access point with an attached MEC server, where the transient segment holding function is deployed as a VNF. More specifically, it is implemented by customizing and extending a Jetty HTTP/2 web server in Java, which realizes all functionalities that are described in Section 3.2.
The MEC server is connected to a satellite terminal that is located at iDirect’s premise in Ireland. The satellite terminal communicates with a satellite gateway (SES Teleport at Betzdorf, Luxembourg) through SES’s owned and operated in-orbit geostationary satellite ASTRA 2F (28.2°E) [13] that operates at Ku-band, and the channel bandwidths are 26MHz and 6MHz for downlink and uplink respectively. This leads to a maximum backhaul TCP throughput of around 60Mbps. The satellite channel is reserved for our experiments during this period. The satellite gateway is directly connected to the 5G core network that is hosted at 5G Innovation Centre (5GIC), University of Surrey, UK. Note that such a topology is due to the facility availability at SaT5G project partners. All experiments were conducted on 24th and 25th September 2018 under clear-sky conditions. A Nokia OZO+ camera is also deployed at 5GIC which outputs 360° monoscopic 4K video at 30FPS and around 3Gbps. The raw video feed is then compressed at a local Matrox Maevex 6100 encoder card, where it is encoded into two RTSP streams with bitrates of 10Mbps and 20Mbps respectively. Constant Bitrate (CBR) is used on both streams and keyframes are inserted every one second. At the live stream origin, each RTSP stream is packaged into three HLS streams with segment lengths of 2s, 5s and 10s respectively, which creates a total of 6 stream scenarios. The HLS streams are delivered via HTTPS to the hls.js client above using TCP as the underlying protocol, which employs BBR as its congestion control mechanism. All streaming sessions are terminated after five minutes worth of video have been streamed.

3.3.2 Performance Metrics

In this work, we evaluate the following QoE metrics:

- Initial startup delay: the duration a client spends waiting before the video starts streaming
- Buffering: given the same 5-minute streaming duration, how long does the streaming stall
- Live stream latency: the gap between the client’s streaming progress and the live origin’s production progress

Under each of the 6 stream scenarios, we begin by evaluating hold-0 scheme’s performance. Hold-0 effectively means that the MEC server plainly breaks the E2E connection into two parts (i.e., RAN and backhaul) and does not hold any video segment’s availability from the client. With the 2-part E2E connection, we then increase the number of held segments (denoted by hold-x) and evaluate their performances accordingly. As x increases, the optimal x is determined if it meets the following criteria:

- The client streamed for 5 minutes without experiencing any stalling.
- The minimal amount of live stream latency is introduced while meeting criteria 1).

Besides the QoE metrics, we also evaluate download throughputs that are experienced by the

![Figure 3-5 Satellite backhaul throughput: all streaming scenarios with hold-0 scheme](image)
client and the MEC server respectively. The former indicates how many segments are successfully downloaded by the MEC server beforehand, and the latter provides more insight into the satellite backhaul link’s TCP performance under various scenarios. Both metrics are measured on a per-video-segment basis. Note that in this work, we do not evaluate the E2E scenario where the client directly streams video from the live origin without passing through any MEC server. This is because it is already verified in [5] that E2E streaming always lead to significantly higher initial startup delay due to TCP’s slow-start and retransmission mechanisms.

3.3.3 Satellite Backhaul Throughput Performance

We begin by evaluating the satellite backhaul’s throughput performance, which indicates the average data rate that the MEC server experienced when it downloads each video segment from the live origin over the satellite link.

We first evaluate the effect of video segment size (in bytes) and length (in seconds) on backhaul throughput. In Figure 3-5, we plot the 95-percentile backhaul throughput results under the hold-0 scheme. Note from the x-axis that we sort the streaming scenarios in ascending orders of segment size and segment length respectively. It is directly observed from that larger segment size leads to higher backhaul throughput, which verifies that file size is the bottleneck of TCP performance over the satellite link. Furthermore, if we compare each pair of streaming scenarios with similar segment sizes, the one with shorter segment length always exhibit higher variance. For example, 20Mbps-2s scenario’s throughput has standard deviation of 0.69Mbps, while 10Mbps-5s scenario’s standard deviation is only 0.07Mbps. This is because shorter segment size means more frequent requests and hence more bursty data transmissions, which is more prone to the satellite channel fluctuation in terms of latency and packet errors. During our experiments, out of the 27,591,951 TCP packets that we sent via the satellite backhaul, there were 133,930 TCP retransmission events which accounts for 0.49% of all packets. This is a significant value in terms of TCP retransmissions and verifies our statement above.

We then evaluate the effect of transient segment holding operation on backhaul throughput. In Figure 3-6, within each streaming scenario, we plot the 95-percentile backhaul throughput results under each holding scheme. First, it is observed that hold-0 and hold-1 schemes experience similar backhaul throughputs, as they both utilize just one TCP connection over the satellite link. Second, it is shown that as more parallel TCP connections are opened, each TCP connection experiences lower throughput in general. This verifies our earlier statement in Section 3.2 that in a channel with limited bandwidth (e.g., satellite link), the number of parallel TCP connections should be limited to avoid causing congestion over the channel.

It is worth noting that as more parallel TCP connections are opened, even though each connection experiences slightly degraded performance, the overall throughput that is aggregated over all connections still increases significantly. This is because the parallel TCP connections are used to download multiple video segments simultaneously. Take the 10Mbps-2s scenario as an example, the hold-0 to hold-5 schemes’ aggregated throughputs’ medians are 2.76Mbps, 2.75Mbps, 5.32Mbps, 7.68Mbps, 10.12Mbps and 12.65Mbps respectively.

3.3.4 Client-Perceived Throughput Performance

We now evaluate the throughput that is actually perceived by the client, i.e., the data rate that the client experiences when downloading each video segment. Recall from Section 3.1 that for a segment download session, its client-perceived throughput takes into account both backhaul throughput (if the segment was not downloaded in time by the MEC server before being requested) and RAN throughput. The 95-percentile results under each streaming scenario and each holding scheme are plotted in Figure 3-7.
First, it is observed that under all streaming scenarios, hold-0 and hold-1 schemes produced very similar client-perceived throughput results, which match their performance over the backhaul. This shows that holding one segment over the satellite backhaul provides little benefit for the client, as the throughput over a single TCP connection is significantly lower than the required video bitrate. As the number of held segments increases, in general the overall client-perceived throughput increases as a higher proportion of video segments are downloaded to MEC server beforehand.

Recall from Section 3.3.3 and Figure 3-6 that in some scenarios, although holding too many segments can cause each TCP connection’s throughput to decrease, the overall throughput that is aggregated over all connections still increases. This statement is further verified in the client-perceived throughput results. Take the 20Mbps-5s stream as an example, the hold-3 and hold-4 schemes produced median backhaul (per-connection) throughputs of 7.6Mbps and 6.8Mbps respectively, which is shown in Figure 3-6(e). However, their aggregated backhaul throughputs are 22.8Mbps and 27.2Mbps respectively, which led to their median client-perceived throughputs of 61.4Mbps and 187Mbps respectively as reflected in Figure 3-7(e). This verifies that despite the trade-off between per-connection performance and the number of held segments, the overall benefit that is brought by utilizing parallel TCP connections to boost aggregated backhaul throughput is still significant.

3.3.5 Client QoE Performance

After evaluating the throughput performance and discussing their patterns, we now assess the clients’ QoE results while focusing on the KPIs that are listed in Section 4.2, i.e., initial delay, buffering duration and live streaming latency. We list detailed statistical results in Table 1 Statistics on key performance metrics’ results. Note that in the table, we also include each scenario’s mean client-perceived throughput and the percentage of video segments that are successfully downloaded by MEC server before they are requested by clients. We believe these two metrics provide additional insights into the QoE performance evaluation. For each streaming scenario in Table 1 Statistics on key performance metrics’ results, we highlight two columns due to their significance. The first highlighted column contains hold-0 scheme that serves as the performance benchmark, and the second highlighted column contains the scheme that holds the optimal number of segments for the corresponding streaming scenario.
The first observation is that under all scenarios, all holding schemes produced almost the same initial startup delay (mostly 2.6s to 2.7s). The initial delay is calculated at the client side using the first request's timestamp and the time that the first video segment is decoded successfully. This observation verifies that breaking E2E content delivery path into two segments at the MEC server can assure initial startup delay, as earlier works showed that video streams often experience initial delay of more than 10s to 20s when being consumed E2E [5]. Since an initial delay of 10s or 20s will cause around 15% or 35% of users to quit watching a stream before it even starts [14], being able to maintain a sub-3s initial delay provides an important assurance for live streaming users’ QoE.

We now look into the 10Mbps-2s scenario’s results. First, both hold-0 and hold-1 schemes produced a mean client-perceived throughput of only 3.1Mbps, which is significantly lower than the required 10Mbps. As a result, both schemes caused the streaming to stall for a total of over 10 minutes (644.8s and 641.2s respectively) while watching 5-minute worth of video. We identify hold-4 to be the optimal scheme in this case, because it is the scheme that produces zero buffering as well as minimal live streaming latency. Although hold-5 produces a higher mean throughput (135.7Mbps compared to hold-4’s 97.3Mbps) and is able to pre-download 84.5% (compared to hold-4’s 53.7%) of all video segments, it incurs a higher live stream latency as well due to the extra segment that is transiently held by the MEC server.

![Graphs showing throughput for different scenarios](image)

**Figure 3-7 Client-perceived throughput: all streaming and holding scenarios, where H0 refers to hold-0 scheme etc.**

We now look into the 10Mbps-2s scenario’s results. First, both hold-0 and hold-1 schemes produced a mean client-perceived throughput of only 3.1Mbps, which is significantly lower than the required 10Mbps. As a result, both schemes caused the streaming to stall for a total of over 10 minutes (644.8s and 641.2s respectively) while watching 5-minute worth of video. We identify hold-4 to be the optimal scheme in this case, because it is the scheme that produces zero buffering as well as minimal live streaming latency. Although hold-5 produces a higher mean throughput (135.7Mbps compared to hold-4’s 97.3Mbps) and is able to pre-download 84.5% (compared to hold-4’s 53.7%) of all video segments, it incurs a higher live stream latency as well due to the extra segment that is transiently held by the MEC server.

<table>
<thead>
<tr>
<th></th>
<th>Hold-0</th>
<th>Hold-1</th>
<th>Hold-2</th>
<th>Hold-3</th>
<th>Hold-4</th>
<th>Hold-5</th>
<th>Hold-6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>10Mbps 2s</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client-perceived Throughput (Mbps)</td>
<td>3.1</td>
<td>3.1</td>
<td>17.8</td>
<td>40.8</td>
<td>97.3</td>
<td>135.7</td>
<td>-</td>
</tr>
<tr>
<td>Total Backhaul Throughput (Mbps)</td>
<td>3.2</td>
<td>3.2</td>
<td>6.1</td>
<td>8.7</td>
<td>11.3</td>
<td>14.1</td>
<td>-</td>
</tr>
<tr>
<td>Initial Delay (s)</td>
<td>2.7</td>
<td>2.7</td>
<td>2.6</td>
<td>2.6</td>
<td>2.6</td>
<td>2.6</td>
<td>-</td>
</tr>
<tr>
<td>Buffering Duration (s)</td>
<td>644.8</td>
<td>641.2</td>
<td>201.6</td>
<td>52.7</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Live Streaming Latency(s)</td>
<td>644.8</td>
<td>643.2</td>
<td>205.6</td>
<td>58.7</td>
<td>8</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>Prefetched Segments (%)</td>
<td>0%</td>
<td>0%</td>
<td>4.7%</td>
<td>16.8%</td>
<td>53.7%</td>
<td>84.5%</td>
<td>-</td>
</tr>
<tr>
<td><strong>10Mbps 5s</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client-Perceived Throughput (Mbps)</td>
<td>5.9</td>
<td>6.1</td>
<td>106.3</td>
<td>178.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>10Mbps 10s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>Initial</td>
<td>Buffering</td>
<td>Live</td>
<td>Prefetched</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Backhaul</td>
<td>Delay (s)</td>
<td>Duration (s)</td>
<td>Streaming</td>
<td>Segments (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Throughput</td>
<td></td>
<td></td>
<td>Latency(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Mbps)</td>
<td>2.6</td>
<td>197.4</td>
<td>197.4</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.2</td>
<td>2.5</td>
<td>187.2</td>
<td>192.2</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11.4</td>
<td>2.7</td>
<td>0</td>
<td>10</td>
<td>47.5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>17.0</td>
<td>2.8</td>
<td>0</td>
<td>15</td>
<td>88.1%</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10Mbps 2s</td>
<td>Client-Perceived</td>
<td></td>
<td></td>
<td>Prefetched</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Throughput</td>
<td>Throughput (Mbps)</td>
<td>8.7</td>
<td></td>
<td>Segments (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Mbps)</td>
<td>9.0</td>
<td>8.8</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.8</td>
<td>17.0</td>
<td>90.6</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>25.4</td>
<td>2.7</td>
<td>212.7</td>
<td>79.3%</td>
<td>89.7%</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20Mbps 2s</td>
<td>Client-Perceived</td>
<td></td>
<td></td>
<td>Prefetched</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Throughput</td>
<td>Throughput (Mbps)</td>
<td>5.1</td>
<td></td>
<td>Segments (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Mbps)</td>
<td>6.7</td>
<td>5.2</td>
<td>13.9</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>29.3</td>
<td>9.7</td>
<td>18.1</td>
<td>22.5</td>
<td>24.6</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>102.6</td>
<td>13.9</td>
<td>22.5</td>
<td>18.1</td>
<td>24.6</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>126.9</td>
<td>2.6</td>
<td>24.6</td>
<td>22.5</td>
<td>18.1</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>122.3</td>
<td>2.7</td>
<td>2.7</td>
<td>26.2</td>
<td>3.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20Mbps 5s</td>
<td>Client-Perceived</td>
<td></td>
<td></td>
<td>Prefetched</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Throughput</td>
<td>Throughput (Mbps)</td>
<td>8.4</td>
<td></td>
<td>Segments (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Mbps)</td>
<td>8.4</td>
<td>8.6</td>
<td>14.9</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>21.6</td>
<td>8.7</td>
<td>22.0</td>
<td>14.9</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>101.9</td>
<td>19.7</td>
<td>27.2</td>
<td>22.0</td>
<td>14.9</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>153.6</td>
<td>2.7</td>
<td>2.8</td>
<td>12.7</td>
<td>2.6</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20Mbps 10s</td>
<td>Client-Perceived</td>
<td></td>
<td></td>
<td>Prefetched</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Throughput</td>
<td>Throughput (Mbps)</td>
<td>10.1</td>
<td></td>
<td>Segments (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Mbps)</td>
<td>10.6</td>
<td>10.5</td>
<td>51.9</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>51.9</td>
<td>18.4</td>
<td>25.2</td>
<td>51.9</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>198.8</td>
<td>18.4</td>
<td>25.2</td>
<td>51.9</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>196.8</td>
<td>25.2</td>
<td>23.8</td>
<td>25.2</td>
<td>23.8</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Under both 10Mbps-5s and 10Mbps-10s scenarios, only 2 segments need to be held to provide a stalling-free streaming experience. This is because the larger segment file size leads to improved TCP performance, which means less parallel TCP connections are needed to bring the aggregated backhaul throughput over the 10Mbps threshold. Meanwhile, note that under the 10Mbps-2s, 5s and 10s scenarios, their optimal holding schemes incurs live streaming latencies of 10.6s, 12.7s and 22.7s respectively. Therefore, given the same video quality (10Mbps) and QoE assurance (i.e., no buffering), despite the TCP performance benefit that is introduced by longer segment length, using 2s segments still incurs the least amount of live latency.

Under the 20Mbps streaming scenarios, the performance patterns are generally similar to the ones under the 10Mbps scenarios. The 20Mbps-2s, 5s and 10s scenarios require 6, 4 and 3 segments to be held respectively to assure the client's QoE. Meanwhile, they incur live streaming latencies of 14.5s, 22.6s and 32.9s respectively. Therefore, even with the higher bitrate requirement (20Mbps), we still establish the same observation that using 2s segments incurs the lowest live latency while assuring a stalling-free streaming session.

Based on the discussions above, we establish the following key observations and conclusions:

- We have successfully validated the effectiveness of our proposed transient segment holding mechanism regarding the assurance of HTTP live streaming clients’ QoE over a real satellite...
backhaul. Furthermore, we use six typical streaming scenarios in real experiments to provide practical guidelines on how to establish segment holding policies in real networks.

• We have verified that file size (in bytes) is the bottleneck of TCP performance over a satellite backhaul. Furthermore, if multiple streaming scenarios have similar video segment sizes, the ones with longer segment length (in seconds) produces better TCP performance.

• We have verified that as more parallel TCP connections are opened over a satellite backhaul link, each individual connection experiences degraded throughput. However, such degradation does not affect the fact that the aggregated throughput over all connections still increases.

• We have observed that given the same video quality requirement, even though longer segment length (i.e., larger file size) means better performance over single TCP connections, the 2s-segment scenarios still incur the least amount of live streaming latency even when more segments need to be held. This provides valuable insights into practical streaming operations, as we have verified that there is no need to use longer segment length in the hope of improving TCP performance, as it may even further degrade live streaming latency.
4 MEC-enabled DASH Video Adaptation in Multi-Link Environments

4.1 Link selection based on MPEG-DASH SAND information

Dynamic Adaptive Streaming over HTTP (DASH) [15] is a technology for delivering video content over the Internet. In DASH, a video file is encoded in multiple bitrates and resolutions. The video file is then split into segments, where a typical segment size is between one and ten seconds. The DASH segments, together with a Media Presentation Description (MDP) that describes the available video representations and the location of the segments, is distributed using HTTP servers, potentially in a Content Delivery Network (CDN).

Network operators and content providers can enhance the delivery of DASH-based video content using Server and network assisted DASH (SAND). SAND introduces the concept of DASH Aware Network Elements (DANEs), elements with minimum intelligence about DASH and awareness of DASH-formatted objects (e.g., MPDs and DASH segments). The DANE is intended to prioritize, parse, or even modify such objects. SAND-enabled DASH clients exchange messages with DANEs to enhance the reception and delivery of DASH content. This message exchange has been standardized in [16]. DANEs that are in the content delivery path and have the ability to pre-fetch content are known as caching DANEs.

A caching DANE can be highly effective when deployed at the edge of a network. Multi-access Edge Computing (MEC) offers content providers this service, by providing computing resources at the edge of the network. The MEC environment potentially offers low-latency and high bandwidth network access between the DASH clients and caching DANEs.

The following SAND messages regarding network QoS information and cache management are relevant for this document:

- **QoSInformation** – This message allows a DANE to inform DASH clients about the available QoS information (e.g., the guaranteed and maximum bitrate between the DANE and the DASH client). DASH clients can take this information into account when requesting segments.
- **Throughput** – This messages allows a DASH client to have knowledge about throughput characteristics and guarantees on the delivery path from DANE to DASH client.
- **AnticipatedRequests** – This message allows a DASH client to announce to a DANE which segments it is interested in (i.e., which segment the DASH client is likely to request soon).
- **AcceptedAlternatives** – This message allows a DASH client to inform a caching DANE when they request a DASH segment that they are willing to accept other (alternative) segments.
- **ResourceStatus** – This message allows a caching DANE to inform a DASH client about segment availability and caching status of the segments.
- **DeliveredAlternative** – This message allows a DANE to inform a DASH client that it is delivering an alternative segment rather than the requested segment. This message is a response to the AcceptedAlternatives message from the DASH client.
- **MaxRTT** – This message allows DASH clients indicating the DANE the maximum time of the request, from when the request was issued until the request needs to be completely available at the DASH client.

In this document, we propose an approach where a caching DANE can be highly effective when deployed at the edge of a network which has a multilink connection to the 5G core network (Figure 4-1).
4.1.1 **Main flow**

In our approach, the DASH client informs the DANE which segments it is likely to request in the near future. Based on this information, the DANE pre-fetches the video segments using one, or both, of the available links. This allows the DANE to download video segments in parallel (e.g., to increase the overall streaming bitrate), or download video segments over a preferred network link (e.g., to perform network link load balancing or to reduce costs).

The following events describe the DANE function and the message exchange between the DASH client and the DANE:

1. The DASH client initiates a new streaming session by downloading the MDP (media presentation description, i.e. the list of media segments) via the DANE.
2. The DANE forwards the MDP to the DASH client and provides the DASH client with QoS information based on the current network conditions. The DANE uses the QoSInformation message for to provide the DASH client this information.
3. The DASH client requests the first video segment.
4. The DANE forwards the segment requests to the media servers using one of the available network links. The DANE may use QoS information when selecting the network link.
5. While downloading the video segment, the DASH client gives the DANE a heads-up about which video segments it is interested in. The DASH client uses the AnticipatedRequests message to give this heads-up. The AnticipatedRequests message also contains a target time at which the DASH client expects to request the indicated video segment.
6. The DANE schedules the segments requests, taking into account the (estimated) data size of the segment, the target time, alternative segments, and network capabilities and load of each of the available network links.
7. The DANE pre-fetches the video segments according to the schedule while the DASH client requests and downloads the video segments from the DANE. During this process the following interactions may take place:
   - To improve cache-hits, the DANE informs the DASH client about the segments that are available in the cache using the ResourceStatus message.
   - To further improve cache-hits, DASH clients list alternative segments that they are willing to accept by adding the AcceptedAlternatives message to the request. When the DANE delivers an alternative segment from the cache, it sends an DeliveredAlternative message to the client.
   - To help the DASH client select an appropriate representation, the DANE sends QoSInformation messages based on the available bandwidth, combining the available resources of the different network links.

Steps 4 to 7 are repeated periodically.

The DASH client is supposed to send consistent messages to the DANE. This means that the list of accepted alternatives should include the same representations in consecutive when conditions remain the same. On changes (e.g., a resolution change at the client or a drop in battery level), the DASH client may specify a different set of accepted alternatives. With respect to anticipated requests, the list should include a list of consecutive requests, following the requested segment. This list may change when a user seeks to a different point in the video.

4.1.2 **Selecting prefetching candidates**

To be able to provide effective traffic splitting, the DANE has to pre-fetch video segments and cache the segments until the DASH client requests them. The pre-fetching candidates (i.e., the video
segments in the manifest that are suitable for pre-fetching) are a selection of video segments defined in the manifest that satisfy the following two conditions:

1. The DASH client should be able to decode and play the video segment, and
2. The DASH client is likely to request the segment in the near future.

The DANE makes this selection of segments based on information provided by the DASH client using SAND messages: accepted alternatives, and anticipates requests. The list of accepted alternatives defines the spatial interest from the DASH client (i.e., the set of representations that the DASH client is willing to accept). The list anticipated requests defines the temporal interest (i.e., the part of video the DASH client is likely to request) of the DASH client. The set of segments conforming to the spatial and temporal interest of the client, also visualized in Figure 4-2, are candidates for pre-fetching.

![Selecting prefetching candidates](image)

**Figure 4-2: Selecting prefetching candidates**

### 4.1.3 Request proxying and segment caching

The DANE should serve the DASH client with a video segment, a quick as possible, and thus preferably from the local cache. Depending on the pre-fetching and caching status, the DANE may do the following:

- **Forward request**: When the DANE receives a video segment request for a video that has not been requested before, not it has been anticipated (e.g., the first requested segment in a streaming session). In this case, the DANE forwards the segment request and provides the DASH client with segment data as soon as the DANE receives this data from the media server. Note that this may happen in as streaming, meaning that the DANE forwards a portion of the download as soon as it receives this portion. In addition, the DANE may decide to store the video segment for serving it to other DASH clients in the future.

- **Request ahead**: In this case, the DANE anticipated a segment request from a DASH client, and already made a segment request for the same segment (or an accepted alternative) at the media origin. When the video segment data arrives, the DANE answers the request from the DASH client and forwards the data. This mode is especially useful for connections with high bandwidth, but with high latency (e.g., when using a satellite link). The time that normally would be lost when creating a network connection (potentially via HTTPS) and making the segment request can be mitigated by sending the segment request ahead based on anticipated requests from the client.

- **Cache and forward**: This case is similar to ‘Request ahead’, with the difference that the DANE already receives segment data before the DASH client has made the request for this segment. In this case, the DANE caches the data. When the DASH makes the segment request, the cached portion of the data is server from the local cache in the DANE. The remainder of the segment is forwarded via the DANE to the DASH client.

- **Full cache**: In this case the segment is fully pre-fetches by the DANE, based on an anticipated request. The segment is served from the local cache in the DANE when being requested by the DASH client.

### 4.1.4 Traffic splitting mechanism

The DANE splits traffic between the different network links on the basis of segment requests. This means that a full segment is transported using one network link, but multiple segments may be
requested in parallel. To use multiple network links, the DANE may pre-fetch anticipated segments using a second network link, which happens in parallel while delivering the requested segments. An example of splitting traffic based on segment requests and anticipated segments is given in Figure 4-3.

![Figure 4-3: prefetching anticipated segments](image)

With pre-fetching in the DANE, it is likely that a DASH client requests a segment that has already been pre-fetched (i.e., a cached segment). In this case, the DANE can directly serve the cached segment to the DASH client. The DANE may further continue pre-fetching segments based on the information (i.e., the anticipated requests) in the request.

### 4.1.5 Bandwidth adaptation

The DANE has to determine the video bitrate when pre-fetching video segments. The bitrate has to be such that the segment can be pre-fetched within the allowed timeframe (i.e., before the timepoint provided by the DASH client in the anticipated requests SAND message). This means that the DANE has to determine the expected throughput for each of the network links.

The DANE can dynamically obtain the throughput on the network links, using network level metrics (e.g., obtained through metering traffic in SDN-enabled switches). Based on the current throughput and network characteristics, the DANE can make a prediction on future throughput. When these aren’t available, the DANE will use download speed measurements of previous segment downloads. As such, the DANE can use an adaptation algorithm for the video bitrate that is normally found in DASH clients.

Note that different network links may require different methods for obtaining the throughput, and the DANE can use different methods at the same time. Depending on competing traffic from other segments downloads by the DANE, the DANE may choose a video bitrate on the basis of the estimated throughput and already active transfers.

Besides using the throughput information for determining the bitrate of a video segment, the DANE may provide this information to the DASH client via a QoSInformation SAND message.

### 4.1.6 DANE architecture

From an architectural perspective, the DANE consists of two modules: (1) a module for communicating with the DASH client and determining which segments (anticipated and alternative segments) should be pre-fetched for this client, and (2) a module for selecting one, or more, network links and do the actual pre-fetching.

![Figure 4-4: DANE architecture](image)

### 4.2 Link selection for SVC Video based on clients buffer level
In this work, we proposed a content delivery strategy which can optimally distribute traffic among terrestrial and satellite backhaul links, without compromising with the video viewing experience of the end-users. Figure 4-5 depicts the block diagram of the proposed framework. To achieve this, the client-side adaptation strategy first selects an enhancement layer for each video segment based on the estimated network condition. Then, SVC-HAS client sends a download request to Multi-access Edge Computing (MEC) [3] [4] [17]. The Link Selection Module (LSM), which is available at MEC, selects a backhaul link (satellite or terrestrial) for each enhancement layer download based on the buffer status of the corresponding client. LSM downloads the enhancement layers of clients having relatively high playout buffer size, through the satellite link. However, the enhancement layer corresponding to clients having critical playout buffer size is prioritized to download from terrestrial link to avoid rebuffering. To realise this, each active client periodically feedbacks its playout buffer status to LSM. Once the layer is downloaded, it is forwarded by the MEC server to the corresponding client. The detailed description of the SVC-HAS client and the MEC server components is given in the next subsections.

![Figure 4-5 Content Delivery Framework](image)

### 4.2.1 SVC-HAS Client

The objective of the SVC-HAS client is to deliver a seamless video viewing experience to the end-user by downloading the video segment at an appropriate enhancement layer. The component of the SVC-HAS client are as follows:

#### 4.2.1.1 Adaptation Module

The adaptation module dynamically selects an enhancement layer for each segment based on the playout buffer criticality and estimated throughput. Throughput estimation plays a crucial role in an efficient layer selection. This work estimates the throughput for the \( s^{th} \) segment download \( \hat{T}(s) \) using the following equation [18]:

\[
\hat{T}(s) = \beta \times \hat{T}(s - 1) + (1 - \beta) \times T(s - 1)
\]

where, \( \hat{T}(s - 1) \) and \( T(s - 1) \) are the estimated and actual throughput achieved by the client for the \((s - 1)^{th}\) segment and \( \beta \) is a positive constant which controls throughput estimation smoothness.

During the initial phase of the video session, the client has little or no information about the download throughput. Therefore, the client takes a conservative approach and downloads the first segment at the base layer. Otherwise, the adaptation strategy selects an enhancement layer \( L(s) \) for the \( s^{th} \) segment as follows:

\[
L(s) = \begin{cases} 
BL & \text{if } B(t) < B_c \\
I & \text{if } B(t) \geq B_c \text{ and } R_1 \leq \hat{T}(s) < R_{i+1}
\end{cases}
\]
Here, $B(t)$ denotes the buffer size of the client at time $t$. It may be observed from the above equation that the adaptation strategy selects the base layer for the $s^{th}$ segment if the buffer size is less than a critical threshold $B_c$. This enables client to quickly replenish the buffer during transient intervals of critically low buffer sizes. On the other hand, when the buffer size is greater than $B_c$, the adaptation strategy always selects the highest enhancement layer $l$ for the $s^{th}$ segment whose encoding bit-rate $R_l$ is less than or equal to the estimated through $T(s)$. The selected enhancement layer is then fed as an input to the Download Module.

### 4.2.1.2 Download Module

The Download Module downloads all the enhancement layers selected by the adaptation module. In traditional SVC client, the download of the enhancement layers is done on a sequential basis [19], i.e., the download of the $l^{th}$ enhancement layer starts after the finishing of $(l-1)^{th}$ enhancement layer download. This sequential download of SVC layer may not be practical over high capacity large latency satellite links. This is because, the file size of the SVC enhancement layer is smaller even for HD and UHD videos and therefore, the achieved download throughput is very low due to the large latency network.

![Figure 4-6 Performance of SVC-HAS client over Satellite backhaul link](image)

Figure 4-6 (a) and Figure 4-6 (b) depict the results of an experiment we carried out to understand the trade-off between sequential and parallel enhancement layer download. In this experiment, we have considered three clients requesting Full-HD videos Big Buck Bunny (BBB), Elephants Dream (ED) and Tears of Steel (TOS) at the highest enhancement layer via emulated satellite link. The experiments have been conducted on three different Segment Duration (SD) (i.e. 2 secs, 4 secs and 8 secs) for each video. A detailed description of the experimental setup is given in section 4. It may be observed from the obtained results (shown in Figure 4-6 (a)) that the clients encounter very high rebuffering percentage during the sequential download of enhancement layer over an emulated satellite link in almost all scenarios. On the other hand, the client is able to significantly reduce the rebuffering event during the parallel download of the enhancement layer in all scenarios, as shown in Figure 4-6 (b).

Therefore in this work, the download module sends concurrent HTTP GET requests to the link selection module for all the selected enhancement layers by the adaptation module.

### 4.2.1.3 Multiplexing Module

This Module generates an AVC file corresponding to each video segment by merging the SVC base layer along with all the enhancement layers. It starts merge operation after the finishing of all the requested layers.

However, the client considers a special case when the media player finishes playing of the current segment, and a few enhancement layers along with the base layer of the next segment to play, are downloaded. In this situation, the Multiplexing Module merges the allowed downloaded
layers and generates an AVC file to avoid rebuffering event. The generated AVC file fed as an input to the media player.

4.2.1.4 Media Player
Media player decodes and plays the multiplexed AVC video segment.

4.2.2 MEC Server
The objective of the MEC server is to deliver ensured QoE to all the clients by optimally utilising satellite and terrestrial integrated 5G network. The components of the MEC server are as follows:

4.2.2.1 Clients’ Buffer Module
In the endeavour to deliver seamless viewing experience to all clients, the MEC server priorities clients having lower playout buffer sizes. To enable this, each client periodically feeds back its playout buffer status to the MEC server and the Clients’ Buffer Module keep an update of instantaneous playout buffer of all the clients.

4.2.2.2 Link Selection Module
Link Selection Module (LSM) attempts to optimise video content distribution among the two available backhaul links (satellite and terrestrial) without compromising with the video viewing experience of the end-users. LSM is responsible for selecting a link for each enhancement layer download during VoD sessions of all clients.

It may be observed from Figure 4-6 (b) that even the concurrent download of layers over the satellite link is not able to restrict the rebuffering, especially for the video with smaller segment duration. This is mainly due to the long latency of the satellite link. Therefore, 100 percentage traffic offloading from the terrestrial network to the satellite link may degrade video viewing experience of the end-user. In an endeavor to deliver seamless viewing experience, the Link Selection Module selects the link based on two factors: (i) The criticality of the client’s playout buffer and (ii) Layer importance in the multiplexing procedure at the client (for example, the base layer is more critical than the enhancement layer in the merge operation).

Figure 4-7 Client playout buffer State

LSM divides all the active clients into three different states (namely, startup, critical and stable) as shown in Figure 4-7. More specifically,

- Startup State: A client is in startup state during the initial phase of the video session. Once the playout buffer size is greater than the threshold $T_{h_{c}}$ client moves from startup state to the critical state.
- Critical State: A Client is considered in the critical state when its buffer size is less than the threshold $T_{h_{c}}$. The chances of rebuffering are high during the critical state of the client.
- Stable State: A Client is in the stable state when its playout buffer size is greater than or equal to the threshold $T_{h_{c}}$. In this phase, the probability of rebuffering is comparatively low.

After the assignment of state to all the active clients, LSM proceeds with the link selection for each enhancement layer download request. LSM selects the terrestrial link to download all layers of the startup/critical state clients. This decision provisions the client to achieve low startup delay during its startup phase and quick replenishment of playout buffer when it is in the critical state.
hand, LSM attempts to offload traffic from terrestrial network to the satellite link for the stable state clients. In order to achieve this, LSM selects satellite link for all enhancement layers of the stable state clients. However, the terrestrial link is selected for the base layer of the stable state client to guarantee seamless playing at least at the lowest quality.

### 4.2.2.3 Layer Download Module
The Layer Download Module downloads the enhancement layer through HTTP connection via the selected link. Once the layer is downloaded, it is forwarded by the MEC server to the corresponding client.

<table>
<thead>
<tr>
<th></th>
<th>BBB</th>
<th>ED</th>
<th>TOS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BL</strong></td>
<td>4.019</td>
<td>4.529</td>
<td>3.813</td>
</tr>
<tr>
<td><strong>EL₁</strong></td>
<td>5.600</td>
<td>6.092</td>
<td>5.417</td>
</tr>
<tr>
<td><strong>EL₂</strong></td>
<td>7.152</td>
<td>7.691</td>
<td>7.023</td>
</tr>
<tr>
<td><strong>EL₃</strong></td>
<td>11.01</td>
<td>12.12</td>
<td>10.98</td>
</tr>
</tbody>
</table>

### 4.2.3 Experiments and Results
The performance of the proposed content delivery framework has been experimentally evaluated using an emulated testbed for satellite and terrestrial integrated 5G network (SaT5G) which is hosted by 5G Innovation Centre (5GIC) at the University of Surrey, UK. Figure 4-8 pictorially depicts the overall network topology used for the experimental setup. As shown in the figure, we have considered a multi-user scenario, having three active clients where each client requests a distinct segmented video file namely, Big Bug Bunny (BBB), Elephants Dream (ED) and Tears of Steel (TOS). In order to emulate satellite and terrestrial integrated 5G, two links between MEC server and the 5G core has been designed. The round trip latency and capacity of emulated terrestrial VPN link are $\sim 25$ ms and 50 Mbps, respectively. On the other hand, latency and capacity of emulated satellite link have are $\sim 550$ ms and 20 Mbps, respectively.

The SVC segments for the considered videos are generated using the JSVM [20] encoder and the demultiplexing tool [19] with three different segment duration (SD) (i.e. SD = 2 secs, SD = 4 secs and SD = 8 secs) and four encoding bit-rates (base layer and three enhancement layer). A summary of encoding bit-rates (measured in Mbps) for each video is shown in Table 2. At each client, a modified version of the DASH-SVC-Toolchain [19] media player has been installed. This toolchain encodes SVC video segments and plays the encoded segments using Mplayer.
4.2.3.1 Performance Metrics

Four quality metrics have been considered for the purpose of performance evaluation and comparison. A description of these metrics is as follows:

- **Startup Delay**: is the total time duration taken by the client to download the first video segment. Startup delay is measured in secs.
- **Rebuffering percentage**: is defined as the total percentage of time the client experiences rebuffering events during its entire video session.
- **Avg Downloaded EL**: This parameter provides an estimate of the mean enhancement level at which the client is able to download its each video segment.
- **Offloading Percentage**: is defined as the total percentage of traffic offloaded from terrestrial link to the satellite link over whole experimental duration.
- **Avg Throughput**: This parameter provides an estimate of the mean throughput achieved by the client for the download of each SVC segment.

4.2.3.2 Results

We now present the experimental results. To measure the efficacy of the proposed strategy, we have conducted two sets of experiments. The values of the constants $\beta_1$, $Th_c$, and $Th_s$ have been taken to be 0.5, 10, and 30, respectively. In the first set of experiments, all the clients have download video segment over satellite link only. On the other hand, both the links have been used by the Link selector module (LSM) in the second set of experiments.

Figure 4-9 Performance of SVC-HAS clients over Satellite link

Figure 4-10 Performance of SVC-HAS clients over satellite and terrestrial integrated 5G network

Figure 4-9 (a) - Figure 4-9 (d) respectively depict the startup delay, rebuffering percentage, Avg downloaded EL and avg throughput achieved by all the three clients when they are streaming video via the satellite link. It may be observed from Figure 4-9 (a) and Figure 4-9 (b) that all the clients encountered high startup delay and rebuffering percentage when they are streaming video over satellite link only. This is because, the download throughput (as shown in Figure 4-9 (c)) achieved by all the clients is very low when the video segments are requested via the satellite link. It may be observed from Figure 4-9 (c) and Figure 4-9 (d) that the throughput achieved by the clients are not even enough to support video streaming at the base layer, especially for the lower segment duration videos. The lower throughput achieved by the client over satellite link is mainly due to two reasons: (i) long latency of the satellite link and thus, HAS performance is bad, and (ii) small file size of the SVC enhancement layer. It may be observed from the above figures that the performance of the SVC-HAS clients is better with higher segment duration due to larger file size. For example, all the clients experienced less than 5% rebuffering when they are requesting videos of segment duration 8secs.

Figure 4-10 (a) - Figure 4-10 (d) respectively depict the startup delay, rebuffering percentage, Avg downloaded EL and avg throughput achieved by all the three clients when they are streaming
video over the satellite and terrestrial integrated 5G network. It may be observed from Figure 4-10 (a) that the client experienced significantly lower startup-delay when it is transmitted through satellite and terrestrial integrated 5G network. This is because, the proposed LSM selects terrestrial link for all layers download during the initial phase (startup state) of the video. Further, the proposed framework is able to significantly reduce rebuffering event (less than 0.2% in all considered scenarios) during the entire video session, as shown in Figure 4-10 (b). The improved performance achieved by the SVC-HAS client is mainly due to the classification of clients in various states and appropriate offloading of traffic from the terrestrial link to the satellite link. Additionally, it may be observed from Figure 4-10 (c) and Figure 4-10 (d) that the proposed link selection and adaptation strategy also help clients to achieve high download average throughput as well as higher avg downloaded EL. Therefore, the proposed link selection strategy enables the framework to offload traffic from terrestrial to satellite without compromising with the delivered QoE to the end-users.

Table 3 Offloading percentage and avg buffer size with varying $th_s$

<table>
<thead>
<tr>
<th>$th_s$</th>
<th>2 secs</th>
<th>4 secs</th>
<th>8 secs</th>
<th>2secs</th>
<th>4secs</th>
<th>8secs</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.05</td>
<td>34.50</td>
<td>51.44</td>
<td>12.07</td>
<td>15.56</td>
<td>92.26</td>
<td></td>
</tr>
<tr>
<td>16.24</td>
<td>34.35</td>
<td>46.89</td>
<td>30.70</td>
<td>34.49</td>
<td>107.73</td>
<td></td>
</tr>
<tr>
<td>15.03</td>
<td>30.08</td>
<td>44.64</td>
<td>57.70</td>
<td>59.94</td>
<td>116.80</td>
<td></td>
</tr>
<tr>
<td>12.46</td>
<td>24.68</td>
<td>34.61</td>
<td>100.56</td>
<td>103.40</td>
<td>146.40</td>
<td></td>
</tr>
</tbody>
</table>

Table 3 Offloading percentage and avg buffer size with varying $th_s$ shows the results of offloading percentage and avg buffer size (measured in secs) over all clients achieved by the proposed framework at four different values of threshold value ($Th_s = 10,30,60,120$) and three distinct values of segment duration ($SD = 2secs, 4secs, 8secs$). It may be observed from the table that the proposed framework is able to offload traffic from terrestrial link to the satellite link and at the same time maintain avg buffer size for each client above or around the threshold value ($Th_s$). The offloading percentage increases with increasing value of segment duration and with decreasing value of $Th_s$. This may be attributed to the fact the client achieves higher throughput with larger segment duration over the satellite link. On the other hand, when the value of $Th_s$ is higher, the framework comparatively transmits larger number of segments/layers through the terrestrial link for the maintenance of higher buffer size at the client. In the literature, it is suggested that a maintenance of buffer size of around 30 secs is able to significantly restrict the rebuffering event even during varying RAN (Radio Access Network) channel conditions [4] [21]. The proposed framework is able to maintain playout buffer size above or around 30, when the selected $Th_s$ value is equal to 30.

### 4.3 MEC based Video-segment Scheduling Network Function

In this work, we have proposed a Video-segment Scheduling Network Function (VSNF), which is deployed at MEC. The proposed strategy has better control over available network resources corresponding to satellite and terrestrial backhaul and, thus, able to schedule segment over both the link more efficiently.

VSNF attempts to deliver enhanced quality of video viewing experience to all clients. Figure 4-11 depicts an abstract overview of the proposed VSNF. The User equipment sends video segment download requests at a particular bitrate to the MEC server, and the proposed VSNF (available at MEC server) handles all the requests during the entire video session. It may be observed from Fig. 2 that the proposed architecture operates using four components namely, Request Handler, Adaptation Module, Link Selection Module, and Enforcer. All the components cooperatively work in unison to deliver satisfactory QoE to each active UE.

From the UE’s perspective, VSNF (which is available at the MEC server) is responsible for handling all the video segment download requests during the VoD session. Each UE sends segment download request to the Request Handler. The requested segment is served immediately to the respective UE if it is available at the MEC server. Otherwise, the Request Handler forwards the segment download
request to the Link Selector Module at a particular bit-rate selected by the Adaptation Module. The
Adaptation Module selects an appropriate bit-rate for each UE based on the estimated SaT5G
backhaul capacity. VSNF invokes the Adaptation Module at the boundary of an adaptation interval.
Typically, the duration of the adaption interval is the order of a few seconds. The Request Handler
receives UE’s segment download request. The Link Selector Module selects a link (satellite or
terrestrial) for the download of each segment. Finally, Enforcer downloads video segment via the
selected link. Description about each of the component is provided in the next subsection.

![Diagram of Video-segment Scheduling Network Function](image)

**Figure 4-11 Video-segment Scheduling Network Function**

### 4.3.1 Request Handler:

Request Handler is responsible for handling the video download requests from all the active UEs. The
requested segment is served immediately to the respective UE if it is available at the MEC server.
Otherwise, the Request Handler forwards the segment download request to the Link Selector Module
at a particular bit-rate selected by the Adaptation Module.

### 4.3.2 Adaptation Module:

The Adaptation Module selects an appropriate bit-rate / enhancement level for each client such that
guaranteed QoE is delivered to each UE, as well as QoE fairness, is maintained SaT5G network, and
(ii) An overall QoE fairness is maintained among flows. It may be noted that such a quality
maximization process must be carried out while ensuring that the total bandwidth demand over all the
flows remains within the expected SaT5G backhaul capacity. In order to achieve this, the adaptation
module first calculates the bit-rate demands ($b_i$) for the $i^{th}$ video flow at the $l^{th}$ quality level based on its
encoding bitrate ($e_i$) and the buffer-filling rate ($BR_i$).

$$b_i = e_i \times BR_i$$

**Problem Definition:**
Let us assume that the available capacity of the SaT5G network is $C$ Mbps. This capacity needs to be distributed among all the $N$ video flows at a given adaption interval boundary. Let $b_{il}$ denotes the bit-rate demand of the $i$th flow at the $l$th quality level with its buffer filling rate. Also, let $x_{il}$ be a binary variable, which is equal to 1 if the $i$th flow is selected at the $l$th enhancement layer. We then formulate the

$$\text{Maximize} \sum_{i=0}^{N-1} \sum_{l=0}^{L-1} (b_{il} \times x_{il})$$

subject to:

$$\sum_{i=0}^{N-1} \sum_{l=0}^{L-1} (b_{il} \times x_{il}) < C$$

$$\sum_{l=0}^{L-1} (x_{il} \leq 1), \quad x_{il} \in \{0,1\}, \forall i$$

The first constraint in the above equation guarantees that the total bit-rate demand of all flows does not surpass the available capacity ($C$) of the SaT5G network. The second constraint forces each flow to select at most one bitrate. The objective of Eq. 6 is to maximize the aggregate video quality for a given limited available capacity.

**Bit-rate/layer allocation:**

In the above subsection, we formulated the bit-rate selection strategy as an Integer Linear Programming (ILP) problem. This bit-rate selection has to be conducted on-line at every adaptation interval. An effective scheduling strategy should provide good solutions quickly. However, it is well known that although an ILP provides optimal solutions, it is inherently exponential in nature and is poor in terms of scalability. The above optimization problem can be solved through branch and bound solution. However, the computational complexity of this solution is $O(NL)$, where $N$ and $L$ respectively represent the total number of active video flow and the total number of bit-rate available for each flow. A closer look into the scheduling problem reveals that the optimization problem can be transformed into discrete by discretizing the capacity constraint. Additionally, the problem has optimal substructure. Hence, the optimal solution may be obtained as a composition of the optimal solutions to a set of its sub-problems. Therefore, Dynamic Programming (DP) provides a natural solution mechanism. In this work we have applied dynamic programming to select an enhancement layer for each client.

### 4.3.3 Link Selection Module

Link Selection Module (LSM) attempts to optimise video content distribution among the two available backhaul links (satellite and terrestrial) without compromising with the video viewing experience of the end-users. LSM is responsible for selecting a link for each video segment download during VoD sessions of all clients. When the buffer size of a client is comparatively lower, then LSM downloads its segments via the terrestrial link to quickly ramp-up the playout buffer. The LSM selects the satellite link to download segment once the buffer size of the client is larger. In order to achieve this, LSM first divides all the active clients into four different states:

- **Startup State:** A client is in startup state during the initial phase of the video session. Once the playout buffer size is greater than the threshold $Th_c$, client moves from startup state to the critical state. In an attempt to minimise startup delay, the link selection module downloads video at the base layer and selects the terrestrial link to download the layers.

- **Critical State:** The client is considered in the critical state when its buffer size is less than the threshold $Th_c$. The chances of rebuffering are high during the critical state of the client. The link selection module terrestrial link to download all the layer of clients which are in Critical state.
- **Transient State**: A client is considered in the transient state if its buffer size is greater than or equal to \( T_h \). For the transient state clients, the base layer is downloaded via the terrestrial link and all the enhancement layers are downloaded via the satellite link.

- **Stable State**: A client is in the stable state when its playout buffer size is greater than or equal to the threshold \( T_s \). In this phase, the probability of rebuffering is comparatively low. All the layer of stable state clients is downloaded via the satellite link.

### 4.3.4 Enforcer

Enforcer downloads the enhancement layer through HTTP connection via the selected link. Once the layer is downloaded, it is forwarded by the MEC server to the corresponding client.

### 4.3.5 Experiments and Results

The performance of the proposed content delivery framework has been experimentally evaluated using a real testbed network for Satellite and Terrestrial integrated 5G networks (SaT5G). Figure 4-12 pictorially depicts the overall network topology used for the experimental setup. In this experiment setup, three clients have been considered to create multi-client scenarios. In each client, an SVC player is installed, and it is requesting a segmented layered video file. All the clients are connected to 5G gNB using CPE. The connection between the client and the MEC server has been established using gNB and a Switch. MEC is a connected CDN server using two backhaul links. In order to create a satellite backhaul, a Satellite terminal along with x7 EC modem is installed by iDirect at the roof of the 5G Innovation Centre (University of Surrey) Building and an iDirect Satellite Gateway is installed at Goonhilly. A two-way connection has been established between Satellite Terminal and Satellite Gateway via Avanti Hylas-4 GEO Satellite. The Satellite Gateway is then connected to the iDirect Satellite RAN, and iDirect Sat Core Network at the Goonhilly. Further, Sat Core network is connected to 5G core (available at the 5G Innovation Centre, University of Surrey, UK) via an L2 VPN connection. The 5G core is connected to the CDN. In order to create a low capacity backhaul terrestrial link, we have created a tunable emulated relationship between MEC and CDN via a switch. A detailed description of the network topology used for this work is provided in Deliverable D.5.3.

Similar to Previous work (Section 4.2), the SVC segments for the considered videos are generated using the JSVM [20] encoder and the demultiplexing tool [19] with three different segment duration (SD) (i.e. SD = 2 secs, SD= 4 secs and SD= 8 secs) and four encoding bit-rates (base layer and three enhancement layer). A summary of encoding bit-rates (measured in Mbps) for each video is shown in Table 2. At each client, a modified version of the DASH-SVC-Toolchain [19] media player has been installed. This toolchain encodes SVC video segments and plays the encoded segments using Mplayer.

In order to measure the performance of the proposed strategy, we have used the same set of performance metric which is defined in 4.2.3.1.
4.3.5.1 Results
We now present the experimental results. In all the experiments, the values of constants $Th_s$, $Th_c$, $Th_t$, and $Th_e$ have been taken as 5, 10, 30 and 45 secs. Figure 4-13 (a), (b), and (c) respectively represent the startup delay, rebuffering percentage, and Avg Enhancement EL. It may be observed from Figure 4-13 (a) that the client experienced significantly lower startup delay when it is transmitted through satellite and terrestrial integrated 5G networks. This is because the proposed VSNF strategy selects the terrestrial link for all layers download during the initial phase (startup state) of the video. Further, the proposed framework is able to achieve comparatively lower rebuffering event (less than 1% in all considered scenarios) during the entire video session, as shown in Figure 4-13 (b). The improved performance achieved by the SVC-HAS client is mainly due to the classification of clients in various states and appropriate offloading of traffic from the terrestrial link to the satellite link. Additionally, it may be observed from Figure 4-13 (c) that the proposed Video-segment Scheduling Network Function (VSNF) is able to download high average enhancement layers (>= 2.5) in all the scenarios which is close to the highest quality (3rd enhancement layer) available for the video. This is attributed to the fact the proposed strategy has better control over network resource allocation and link selection for each segment download.

Figure 4-13 Performance achieved by SVC-HAS clients using VSNF strategy
Figure 4-14 depicts the plots for instantaneous offloading percentage achieved by VSNF over the entire experiments duration in three distinct values of segment duration (SD=2secs, 4 secs, and 8secs). For example, the plot shown in the red line represents the scenario in which all the clients are requesting the video of duration 2 secs from the content server. It may be observed from the figure that the proposed framework is able to offload high percentage of traffic to the satellite link in all three scenarios. Further, the offloading percentage achieved by the proposed strategy increases with increasing segment size and reaches above 90% for the SD = 8secs scenario. This may be attributed to the fact that the performance of the satellite link is comparatively good while downloading larger file size.

Figure 4-14 Instantaneous Offloading Percentage
5 Content Caching and Placement Optimisation at Satellite-backhauled Flights

SaT5G Use Case 4 (UC4) revolves around connectivity to moving platforms through the integration of the Satcom in the 5G technology stream. Specifically, the focus is on the aviation vertical with 5G giving rise to the opportunity to focus on the next generation of aircraft connectivity. The typical configuration on-board an aircraft’s cabin is shown in Figure 5-1, where seat screens are connected in Daisy chain fashion to the central media server, while person devices (PEDs) of passengers are connected over the existing Wi-Fi system. As we can notice from the figure, the existing networking has been quite simplified, and it follows ARINC 628 specifications. Referring to Figure 5-2, as 5G technology is becoming available, this gives rise to the possibility to re-design the cabin network infrastructure according to a new paradigm, thus providing 5G to complement Wi-Fi radio access. This is not only limited to the cabin infrastructure, but it extends also to the satellite connection that allows aircrafts to get connectivity to a ground infrastructure.

Based on the idea to evolve the whole connectivity system for aircrafts (both inside and outside the cabin), we consider a situation in which 5G, in the future, can give room to a higher degree of freedom to deploy an aeronautical system for connectivity. Moreover, network virtualisation and programmability can reduce the dependence on dedicated expensive hardware, thus fostering competition between stakeholders, which in turn may result in lowering the costs of the connectivity infrastructure as a whole. Based on this premise, in this work, we envisage a scenario in which content can be uploaded in the cabin infrastructure by means of the satellite system unlike nowadays. Indeed, as of today, media content for passengers is loaded into the central media servers when aircrafts are grounded whereby a time-consuming process (several hours). In addition, the existing content loading procedure restricts to deploy a single media catalogue on-board, which is changed or updated in a relatively long period of time (weeks or months). However, not all media contents that are deployed on-board have the same popularity and, sometimes, volatile content (e.g. videogrip) that is shared in conjunction of specific events may become viral. An example is provided by major sport events or popular events like the Academy awards. In such context, airlines might want to update their media catalogue on certain routes or even create a second catalogue where viral clips with a short timespan can be cached.

To fully lay down the foundations of our approach, we point out that the evolution of the connectivity system on-board aircrafts is envisioned to pass through intermediate steps in which unlicensed bands will still play a key role. Besides Wi-Fi, as an example of technology operating in the 5 GHz ISM band is provided either by LTE Licensed Assisted Access (LTE-LAA) or LTE unlicensed (LTE-U). In view of such a possibility, in line also with the activities carried out in WP5 by the ZII test-bed, we envisaged a virtualised infrastructure on-board in which content can be stored dynamically depending on popularity and offered to travellers through the on-board connectivity system. Therefore, not only matters what content to store on-board but also how quickly this can be accessed by requester (i.e. typically a passenger), and that pertains to bit error probability analysis of the Geostationary satellite link that operates in the Ka band.

It worth mentioning that caching strategies and related problems have received significant attention in the past years. Several research works were developed on this topic modelling the caching problem in different contexts, including the research line of edge computing. Some studies related to caching can be found in [22] [23] [24] [25]. A popular approach to study the benefits of caching in a network infrastructure that avails network virtualisation of the storage and compute resources consists in developing a problem that is solved with optimisation theoretic tools, but that typically turns to be NP-Hard to solve. Common problems that can be found in studies are the placement of the cache(s), decide what content to cache based on a hit rate and content popularity and user association to a wireless transmitter that is the best for providing content from the cache. In complex optimisation, heuristics are developed to reduce complexity and find solutions that can trade-off complexity and utility maximisation.

Objective of the study: demonstrate a caching strategy to load content on-board aircrafts as an application of Multi-access Edge Computing (MEC) use case that 5G features can unlock. In the context of an aircraft, 5G features refer also to the possibility to create a virtualised infrastructure that
is open to different service providers or to airlines to instantiate new services in a more flexible manner compared to nowadays. Content has to be made available to the passengers’ PEDs in a different way compared to nowadays in which passengers mainly use their seat screens on a long-haul flight.

Remark: At this initial stage, it is worth stressing the fact that cellular communications on-board civilian aircrafts are very limited and at mostly up to 3G cells to enable voice communication.

For the sake of clarity, the scenario is shown in Figure 5-3. We shall hence consider the problem divided in two parts. The first part consists of deciding what content has to be cached on-board the aircraft out of the pool of the whole content available on the ground infrastructure. In this first part, we shall assume that the ground infrastructure has an unlimited storage and that any ground network has negligible cost compared to the use of the satellite bandwidth. In the second part of the problem instead, we model what content can be moved closer to the passengers, assuming that additional storage can be made available in the aircraft other than in the central media server. For this second part, we shall assume to use both Wi-Fi technology and LTE technology operating in the unlicensed band. We do not consider the possibility to use the wired network on-board since this is limited to GbE and already in use for more conventional consumption of content through the seat screen devices.

Based on this, we further break the strategic caching problem on-board aircrafts into different problems that are evaluated analytically:

- We model the popularity of contents using a two-state model
- We model the link quality of a satellite connection in case of a Geostationary stationary satellite system, which is de facto the satellite technology that can provide global coverage
- We develop the strategy to cache content on-board (either in the central media server or in a secondary catalogue even)
- We decide which content can be moved as closed as possible to the passengers.

The remainder of this section is organized as follows. In Section 5.1 we present the system model behind our work, including the necessary assumptions. In Section 5.2 we provide in-depth description of the analytical model that we propose, providing also the necessary demonstration. In Section 5.3 we show the numerical results, while in Section Error! Reference source not found. we provide the concluding remarks of the section, highlighting also the main outcomes of the work.

![Figure 5-1: Typical networking configuration for seat screens and personal devices on-board aircrafts](image-url)

Page 45 of 61
5.1 System model

To study caching in the context of an aircraft moving platform, we assume that on the ground a catalogue of media contents is available in a facility where storage capacity can be assumed infinite and networking based on optical fiber so that to be as close as possible to an ideal case in terms of network resources. The media content may belong to an airline company, an operator itself or to a third-party content provider. The integration of satellite connectivity in 5G is expected to lower the costs of the satellite bandwidth by means of fostering the competition between stakeholders, that allows also new joint ventures and in overall it spurs a system that is less dependent from hardware specific choices. As already mentioned, NFV and SDN are the enablers to deploy specialised software on COTS hardware. In this context, a 5G system that targets a fleet of aircrafts can contribute to make more readily available not only connectivity per se but also new services. Therefore, besides the traditional catalogue of content that is currently loaded on-board the central media server during the time aircrafts are grounded (through a time consuming process), a second catalogue that caches more volatile popular content can be created using the satellite connection and the end-to-end 5G system.
In this work, we assume that both on the ground and on-board an aircraft there is the possibility to deploy virtualised services either through a cloud controller such as Openstack or resorting even to lighter weight approaches such as containers. This is mainly a technology dependent choice that positions our study in the proper context without affect it. We assume that a network service orchestrator deployed on the ground can start virtualised services before an aircraft comes into service or even during the flight time. Moreover, we assume that the virtualised 5G mobile core network functions are split between on-board the aircraft and in a ground infrastructure (e.g. operator central office). The mobile core network functionality deployed in the aircraft infrastructure is denoted as MEC hereinafter, whereas the remaining part of the mobile core that is deployed on the ground is simply denoted as core network (CN) in the remainder of this document.

From the radio access standpoint, we assume to enhance the existing Wi-Fi access on-board aircrafts by means of LTE unlicensed eNBs and that both Wi-Fi and LTE make use of the 5 GHz band. Referring to the 5G mobile core network architecture defined in 3GPP Release 15 [26], a Non 3GPP Interworking Function (N3IWF) can ensure the integration of a non 3GPP technology like Wi-Fi in the mobile core network. For LTE operating in the unlicensed band, LAAs or LTE-U could be used, although LTE-LAA has proven better coexistence with Wi-Fi compared to LTE-U [27] [28]. At this point, it is worth stressing the fact that when two OFDM-based technologies that are not coordinated and that use different OFDM numerology have to coexist in the same radio environment interference may arise still for example due to hidden node problems. On the other hand, connectivity to the 5G mobile core network can be ensured for LTE considering core network architectures that allows still using the 5G User Plane Function (UPF) for the data plane and 4G backward compatible signalling.

In this work, we target to demonstrate a caching strategy for the case of an aircraft moving platform within the context previously laid down. As already mentioned, we consider on the ground an unlimited storage capacity whereas instead this is assumed limited on-board an aircraft.

**Problem formulation 1:** we target to decide which content that is available on the ground shall be cached on-board the aircraft, for example in the central media server, taking into account the availability of satellite link bandwidth and the satellite link quality.

**Problem formulation 2:** once content has been cached from the ground to on-board the aircraft, this can be made available to the passengers’ PEDs through the heterogeneous wireless connectivity system on-board (i.e. a mixture of Wi-Fi and LTE). Therefore, we envisage that some PED can be connected to the on-board Wi-Fi network or to the on-board LTE network with both operating in the 5 GHz band. As discussed briefly, we assume that the GbE network on-board is mostly used for the seat screens. In addition, we assume that, in a future aircraft environment, additional storage can be made available on-board where media content can be strategically cached to allow passengers to access it as fast as possible. This will enable an edge moving platform that fully embraces the concept of MEC. An important aspect to decide whether it is worth to cache content closer to the passengers or not is the quality of the wireless network on-board, which can be addressed by studying the interference configuration.

Additional assumptions that we use in our work are the following:

- We assume that a media content is a single unit to be cached and that the overall storage capacity on-board an aircraft is limited but discretized in terms of media content units,
- We shall model the popularity of a content through the two-state model with transition probability $\gamma$ shown in Figure 5-4,
- We denote with $\omega \in [0,1]$ the fraction of the total satellite bandwidth that is made available for the satellite link,
- We assume a Geostationary satellite connection on the Ka-band in which QPSK modulation is used,
- We model the success probability to transfer packets of $L_B$ bits over the satellite connection relying on the work already developed in [29], in which the influence of the atmospheric turbulence is introduced,
- The heterogeneous wireless connectivity system on-board an aircraft is studied from a spatial standpoint whereby using stochastic geometry statistical tool in case one Wi-Fi transmission overlaps with one active LTE downlink communication,
- For LTE, we shall assume a small cell type of device,
- In the remainder, we shall denote with Radio Transmission Point (RTP) a generic radio equipment that connects passengers’ PEDs that can be enabled with both LTE and Wi-Fi.
radio cards, as well as a radio transmitter can onboard either of the two wireless technologies,

- We assume that the active Wi-Fi link uses an $M$-QAM modulation, while the LTE downlink makes use of $M'$-QAM with $M \neq M'$,
- We compute the packet success probability of a packet of $L$ bits that is transmitted by an RTP in case the communications links (both Wi-Fi and LTE) are affected by Rayleigh distributed amplitude fading (exponentially distributed power fading) and Log-Normal shadowing,
- We shall denote with $\alpha$ the path-loss exponent for the wireless links on-board the aircraft,
- We shall model the caching strategy resorting to the statistical tool of urn models, which is particularly suitable for solving combinatorial discrete problems,
- In the discrete system, we assume an indivisible unit of media content to cache and a unit of storage capacity for that purpose,
- The popularity of contents to be cached is modelled through a random variable $g$ that denotes in which state of popularity a content can be either in a popular state or in a non-popular state,
- We denote with $g_c$, and $c \in [0, \infty)$ the independent and identically distributed random variables that model the popularity of contents. Referring to Figure 5-4, each content will have its own transition probability $\gamma_c$, which is assumed the same for all contents in the remainder of this work,
- The urn problem that models the strategy to cache contents available on the ground in the aircraft depends on content popularity and a threshold $b$ that denotes the hit rate of such a content
- The urn model to cache contents close to the passengers’ PEDs is based on a random assignment and on the link quality of the wireless system on-board.

The two-state model mentioned above for the content popularity shown in Figure 5-4 can be straightforwardly solved in closed form as $\Pi_0 = 1 - \gamma$ and $\Pi_1 = \gamma$, where $\gamma$ denotes the transition probability, $\Pi_1$ the steady-state probability for a content to be in the popular state and $\Pi_0$ the vice versa.

$\begin{align*}
0 & \xrightarrow{\gamma} 1 \\
\Pi_0 & \xrightarrow{1-\gamma} \Pi_1
\end{align*}$

Figure 5-4: Two-state model approach to content popularity

Urn models [30] are a powerful tool to study either the single urn case or the multi-urn case. In the multi-urn case, it is worth mentioning the well-known coupon’s collector problem. On the other hand, a single urn case is often study to model virus’s evolution. In our work, we shall use the multi-urn approach with the assignment of $n$ balls into $m$ urns. In the specific context of modelling the caching problem, we shall assume that $m$ urns are discrete units of media contents to cache, while $n$ balls are the discrete media content units that the on-board storage allows to cache.

It well known that urn problems can be highly complex from an analytical standpoint, but a very powerful tool is provided by the exponential generating functions (EGFs), which is one of the many possible mathematical transformations. Generally operating in the transformed domain offers specific advantages such as the possibility to compute in closed form all the moments associated to a random
variable (r.v.) of interest. As such, EGFs are particularly suitable to model different problems for the assignment of \( n \) balls into \( m \) urns in the case which an urn can accept an infinite number of balls. The generic expression \( F(x, y) \) for the EGF in the urn model representation is shown in equation (1).

\[
F(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} h_{i,j} x^i y^j = \prod_{j \in S} (g(y) + (x - 1)f(y))^{m_j},
\]

where we denote with \( S \) the set of possible urn types, with \( x \) the transform variable for urns and with \( y \) the transform variable for balls allocated into the urns. The expression in equation (1) takes into account the fact that non-empty urns (denoted by \( x - 1 \)) can be divided in different category with \( m_j \) denoting the number of urns in the \( j \)-th category. On the other hand, \( g(y) = e^y \) denotes the generic generating function for an urn that can accept an infinite number of balls (or hits in case of caching) and \( f(y) \) instead denotes a specific allocation (or strategy) of interest. In case of content caching, \( f(y) \) is the transformed function of the possible ways of selecting contents. Interestingly, equation (2) allows computing the \( k \)-th moment for the number of urns that fulfils a condition of interest expressed by \( f(y) \), which is computed using the coefficient of the transform variable \( y^v \). In our work, we are mostly interested in computing the average number (1\(^{st} \) moment) of non-empty urns that fulfils conditions of interest.

\[
E\{x^k\} = \left[ y^v \right] \left( \frac{\partial^k}{\partial x^k} F(x, y) \right) \bigg|_{x=1}
\]

**5.2 System analysis**

In this section, we provide the basis for the analytical work that we have developed to model caching in the context of an aircraft moving platform.

![Geometric model to compute the bit error probability over the satellite link](image)

Figure 5-6: geometric model to compute the bit error probability over the satellite link [29]

### 5.2.1 Modelling success probability for the GEO satellite connection

In this section, we present the model for computing the average bit error probability \( p_b^{(sat)} \) over the Geostationary satellite link operating in the Ka-band. As anticipated in Section 5.1, the analysis relies on the work developed in [29]. Particularly, modelling the impact of atmospheric turbulence on the electromagnetic wave that propagates in a random medium. The model proposed in [29] is based on a Gaussian model of the atmospheric turbulence and the average bit error probability is based on the probability density function (PDF) of received intensity, with the function obtained from the moments of the receive intensity. The geometric model behind the work done in [29] is shown Figure 5-6. In the figure, \( R \) denotes the radius of the earth, \( L \) is the altitude of the Geostationary satellite, \( h \) denotes the
altitude from the ground and \( \theta \) the elevation angle of the transmitting antenna. Based on [29], we denote with \( z \) the propagation distance that is computed as

\[
z = \sqrt{(R + L)^2 - (R \cos \theta)^2} - \sqrt{(R + h)^2 - (R \cos \theta)^2}
\]

and the correlation function of the random dielectric constant with

\[
B(r_-, z_+, z_-) = B(z_+) \exp \left[ -\frac{r^2 + z^2}{l^2(z_+)} \right]
\]

With \( B(z_+) \) the local intensity of the random medium and \( l(z_+) \) the correlation length of the random medium. Specifically, the random intensity of the random medium is assumed a function of the altitude as

\[
B(h) = B_a \left( 1 + \frac{h}{h_0} \right)^{-2/3}, \quad \text{for } 0 \leq h < h_1
\]

\[
= B_a \left( 1 + \frac{h_1}{h_0} \right)^{-2/3} \left( \frac{h}{h_1} \right)^{-4/3}, \quad \text{for } h_1 \leq h < h_2
\]

\[
= B_a \left( 1 + \frac{h_1}{h_0} \right)^{-2/3} \left( \frac{h_2}{h_1} \right)^{-4/3} \exp \left( \frac{h - h_2}{h_0} \right), \quad \text{for } h_2 \leq h \leq h_t,
\]

with the altitude \( h \) expressed in kilometres.

Based on the work developed in [29], the bit error probability for the Geostationary satellite link can be written as

\[
p_b^{(sat)} = \frac{1}{2\sqrt{2\pi}\sigma_e} \int_{0}^{\infty} \frac{1}{u} \exp \left[ -\frac{\ln(u) - m_e}{2\sigma_e^2} \right] \text{erfc} \left( \sqrt{\frac{E_b}{N_0}} u \right) du,
\]

considering that the conditional bit error probability for a QPSK modulation is \( p_b(u) = Q(\sqrt{u}) \), where the conditioning is on the distribution of signal-to-noise-ratio per bit \((E_b/N_0)\). In equation (5) the PDF of the signal-to-noise-ratio per bit is considered the one of a Log-Normal distribution with parameters \( m_{E} = -h(z) \), \( \sigma_{E}^2 = 2h(z) \).

The integral shown in equation (5) can be evaluated numerically solving also the following integral numerically

\[
h(z) = -8 \int_{0}^{x} dz_1 \int_{0}^{z-z_1} \frac{z-z_1}{2} \frac{z_2}{2} b_2 \left( z-z_2 - \frac{z_1}{2}, z_1 \right)
\]

with

\[
b_2(r_-, z_+, z_-) = -\frac{[\nabla^2]^{i+1} B(r_-, z_+, z_-)]_{r_-=0}}{2^{2i+1}(i+1)!^2}
\]

and \( \nabla = \frac{i\omega}{\frac{\partial}{\partial x}} + \frac{i\omega}{\frac{\partial}{\partial t}} \).

Finally, we are able to compute the success probability for an uncoded packet of \( L_p \) bits as

\[
P_s^{(sat)} = (1 - p_b)^{L_p}.
\]

Assuming as numerical values that \( B(z_+) = B(h) \) as per equation (4) and \( l(z_+) = 0.05 \text{ km}, h_1=0.05 \text{ km}, h_2=2 \text{ km}, h_3=20 \text{ km}, h_{s1}=0.002 \text{ km}, h_{s2}=1.75 \text{ km} \) and \( B_a=4.5 \times 10^{-9} \), after evaluating numerically the
integral for $h(z)$, we are able to obtain $p_b$ and consequently $p_b^{(sat)}$ as shown respectively in Figure 5-7 and Figure 5-8 as a function $E_b/N_0$. For the sake of obtaining results, the $E_b/N_0$ range was set in the interval between -5 dB and 20 dB and the elevation angle $\theta = \pi / 18$ radians.

Figure 5-7: Bit error probability for a Geostationary satellite link in the Ka-band

Figure 5-8: Packet success probability computed for the Geostationary satellite link
5.2.2 Modelling success probability for an on-board connection

In this section, we develop the bit error probability analysis for the heterogeneous wireless system onboard an aircraft. As mentioned, we assume the presence of both Wi-Fi and LTE operating in the 5 GHz band that can be used to make content available to the passengers after a part of the whole catalogue of content has been available on ground has been cached. We discussed already that LTE in unlicensed bands and Wi-Fi links can cause mutual interference despite detect-and-avoid mechanisms like the Listen-Before-Talk are in place in LTE-LAA and energy detection in Wi-Fi.

Referring to the assumptions stated in Section 5.1, we assume that the two wireless technologies can give rise to mutual interference. Hence, we study the general case of an RTP that delivers downlink traffic to a PED. Referring to Figure 5-9, the bit error probability analysis and the corresponding packet success probability can be developed for a test receiving PED in conditions of interference when one Wi-Fi transmitter and one LTE small cell are active. The analysis developed hereinafter does not distinguish whether the receiving PED under test is connected over LTE or Wi-Fi but rather generally describes interference configurations.

The problem of obtaining bit error probability and packet success probability under general propagation and network assumptions is a topic that was intensively studied in past years. The analysis of stochastic spatial processes found applicability in this field. In this regard, the powerful statistical tool of Stochastic Geometry found applicability in studying interference in wireless networks from the spatial point of view [31]. Some notable examples of applications of Stochastic Geometry to study wireless networks can be found in [32] and [33]. The most widely used assumption in research works that study interference in wireless networks consists of using spatial Poisson Point Processes with constant intensity or, in other words, spatial density. Several research studies have demonstrated the suitability of this assumption to develop quite accurate models in wireless networks, alongside with analytical tractability that allows obtaining closed form expressions.

As discussed already, LTE and Wi-Fi are based of OFDM modulation, although the numerology behind these two physical layers is radically different (e.g. sub-carrier spacing). Anyway, while one technology makes use of $M$-QAM, the other uses $M'$-QAM with $M \neq M'$ generally. Moreover, the analysis is developed assuming Rayleigh distributed amplitude fading (i.e. exponential power fading) for all active wireless links and that they are also affected by Log-Normal shadowing. In the remainder of this work, we shall denote with useful link the wireless communication destined to the test PED receiver and with interfering link the other wireless link based on a different OFDM technology (compared to the useful link) but that is also active. The test receiver is also assumed located at the centre of the two-dimensional plane where the reference system is centred. As such, the random distance ($r$) between the interfering transmitter and the test receiver can be modelled by means of a
r.v. with probability \( f_u = \frac{2u}{u^2} du \) in the elemental interval \([u, u+du]\). For completeness, we shall denote with \( r_u \) the length of the useful link, which is used as parameter that can be varied to obtain numerical results. All distances are bound within the area \( A \) of interest of radius \( R_u \).

The bit error probability \( (p_b) \) for an \( M \)-QAM modulation is provided in high SNR regime by equation (8), which is based on the work developed in [34]:

\[
P_b(\eta) \equiv \frac{4}{\log_2(M)} \times \left( 1 - \frac{1}{\sqrt{M}} \right) \times l_{\eta} - \frac{4}{\log_2(M)} \times \left( 1 - \frac{1}{\sqrt{M}} \right)^2 \times l_{\eta}^2
\]  

(8)

with

\[
\ell_{\eta} = \frac{1}{\pi} \int_0^y \left( 1 + \frac{g_b}{\sin^2 x} \eta \right)^{-1} dx
\]

and

\[
\eta := \frac{e^{2\sigma_I c_s}}{r_u^{\alpha}} (V + 1) \times \epsilon_{bs} \frac{N_0}{N_0}
\]

where \( \epsilon_{bs}/N_0 \) is the signal-to-noise ratio per bit for the useful signal, \( \sigma_I \) the shadowing factor that affects the useful link, \( \sigma_s \) the shadowing factor that affects the interfering link and \( N_0 \) the one-sided AWGN noise power spectral density.

In case of a single Wi-Fi transmitter and a single LTE transmitter that are active, we are able to rewrite \( \eta \) as follows

\[
\eta_1 = \frac{2b}{a} \times \frac{R^\alpha}{2+c} \times 2F_1\left(1,\frac{2+\alpha}{\alpha},2,\frac{-R^\alpha}{a}\right)\Gamma(1+2/\alpha)
\]  

(9)

where \( \Gamma() \) denotes the complete Gamma function and \( 2F_1() \) the hypergeometric function. In addition

\[
a = \frac{2}{3} e^{2\sigma_s} \log_2 M' \times \frac{\epsilon_{b1}}{N_0}
\]

and

\[
b = e^{2\sigma_s} \log_2 M \times \frac{\epsilon_{bs}}{N_0} r_u^{-\alpha}
\]

The result provided in equation (9) can be obtained by considering that the term \( V \) in \( \eta \) can be rewritten as

\[
V = \frac{2}{3} \epsilon_{b1} \times \frac{e^{2\sigma_I}}{r_I^{\alpha}},
\]

where \( \epsilon_{b1}/N_0 \) denotes signal-to-noise ratio per bit in the interfering transmission. This yields the possibility to rewrite the SINR as follows

\[
\eta_1(\eta) = \frac{e^{2\sigma_s} \log_2(M') \times \frac{\epsilon_{bs}}{N_0} r_u^{-\alpha}}{1 + \frac{2}{3} e^{2\sigma_I} \log_2(M') \times \frac{\epsilon_{b1}}{N_0} r_I^{-\alpha}}.
\]  

(10)

Taking into account the assumption that the target receiver under test in located at the centre on a two-dimensional reference system and that nodes are uniformly scattered within the area \( A \) with radius \( R_u \) (see Figure 5-9), we can remove the dependence of in equation (10) from the position of the interferer.
\[ \eta_1 = \int_0^\infty \eta_1(r_i) \frac{2r_i}{R_u^2} dr_i . \]  

(11)

The integral in equation (11) can be solved in closed form using the result of the integral shown below for any complex constant \( G \in \mathbb{C} \) with \( \mathbb{C} \) denoting the set of complex numbers. The integral was obtained with the tool of Mathematica in closed form.

\[ \int_0^{R_u} \frac{1}{1+Gu} \frac{2u}{R_u^2} du = \frac{2R_u^3 \times \text{2F1}\left(1,\frac{2+\alpha}{\alpha} ; 2+\alpha ; \frac{R_u^2}{\alpha}\right)}{\alpha(2+\alpha)} . \]

Based on the analysis shown above, we are able to compute the success probability \( (p_s) \) for an uncoded packet of \( L \) bits. The result of the analysis is shown in Figure 5-10 assuming that both LTE and Wi-Fi communications make use of 20 MHz bandwidth, the transmitted power of Wi-Fi is 20 dBm and that of LTE is 18 dBm. The signal-to-noise-ratio per bit for both useful and interfering communications is thus obtained as a result of a simple link budget computation. The path-loss exponent is assumed \( \alpha=4 \) as for an indoor environment like an airplane. The system operating frequency is 5 GHz, the radius \( R_u \) of the area under study is up to 15m. The shadowing factor for both useful and interfering links is assumed 10 dB.

![Figure 5-10: Packet success probability in a heterogeneous access network made of Wi-Fi and LTE unlicensed](image)

**5.2.3 Urn models for content caching in an aircraft moving platform**

The urn model that we develop for the caching strategy in case of an airplane moving platform in which the end-to-end system is enabled by 5G is explained hereinafter relying on the EGF approach provided in [30]. The basic assumptions behind the system model were explained in Section 5.1 already. In the system model section, it was indeed explained the fact that we model the caching strategy problem with urns and balls, where an urn is an indivisible discrete unit of media content to cache and a ball is a discrete amount of storage where to store the content.

The model for deciding what contents can be cached on-board an aircraft considering the amount of media content units stored on the ground (where unlimited storage and network resources is assumed) is based on a ball-urn random assignment strategy subject to the specific constraints that a content can be selected an infinite number of times as correspondent to the content hit rate (i.e. a common way to measure popularity) and that a content hit rate threshold \( b>0 \) is assumed. A content
popularity can be modelled with a r.v. that is statistically independent from that of other contents so that product forms can be obtained in the analysis.

As mentioned, the caching problem under study is modelled whereby a discrete system in which the variability of the caching contents is assumed to be in a state that is statistically independent from the variability of other contents. In Section 5.1, the set \( S = \{0, 1\} \) is divided in two subsets, the one of popular discrete items and the one of non-popular items with the term 'item' referring to a discrete media content unit.

- The number of urns corresponds to the number of discrete media content units that are in overall available on the ground. The number can be computed as: \( m_j = m_t \times \Pi_j \), where \( m_n \) denotes the total number of discrete unit of contents that can be cached, \( \Pi_j \) is the steady-state probability to be in state \( j \) for any of the \( m_b = m_o + m_n \) contents and \( m_o \) denotes the type of discrete units (i.e. urns) of type \( j \in S \) where \( S \) denotes the set of possible element types. In the case under study, referring to the two-state model illustrated in Section 5.1, the set \( S = \{0, 1\} \) is divided in two subsets, the one of popular discrete items and the one of non-popular items with the term 'item' referring to a discrete media content unit.

- The number of balls in the caching Problem Formulation 1 (see Section 5.1) depends on the storage capability on-board an aircraft that in the first place is assumed to be the central media server. For instance, in case of a total storage capacity on-board denoted by \( \Delta \) (in Gigabytes) and a specific content size \( \delta \) (in Gigabytes), we can get \( n_z = [\Delta/\delta] \) with \([\cdot]\) that denotes the lowest integer number higher that the quotient. Based on \( n_z \), we are able to compute the effective number of balls to use in the model: \( n = n_z \times p_s^{(sat)} \times \omega \times \gamma \). This shows that based on the fraction of available satellite bandwidth, the satellite link quality and the content popularity there will be more or less room to cache content. Intuitively, contents with low popularity have less possibility to be cached.

- In Problem Formulation 2, the total number of contents to be cached in additional distributed storage locations that can be displaced inside the aircraft (either in the RTP or in external servers other than the central media server) can be denoted as \( m_g \), which is the result of the average number of non-empty urns computed in Problem Formulation 1.

The number of balls in the caching Problem Formulation 2 (see Section 5.1) is computed based on the storage capacity \( K \) to cache discrete unit of media contents in distributed storage locations along the aircraft, and is based on the link quality of the wireless connectivity system on-board: \( n_a = [K \times p_s] \).

In Problem Formulation 1, the EGF function that allows to model the caching strategy problem described in Section 5.1 is the following

\[
F(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} h_{ij} x^i y^j = \prod_{j \in S} \left( g(y) + (x - 1)(g(y) - g_b(y)) \right)^{m_j},
\]

with

\[
g(z) = \sum_{i=0}^{\infty} \frac{z^i}{i!} \quad \text{and} \quad g_{r}(x) = \sum_{q=0}^{b} \frac{x^q}{q!} \]

that are obtained from the Taylor expansion series of an exponential function. Based on equation (2), we are able to write

The first moment (or average) for the number of non-empty urn, which corresponds to the discrete units of content to cache on-board:

\[
E\{x\} = m_g = \sum_{j=1}^{S} m_j \left( 1 - \left( 1 - \frac{1}{m_j} \right)^{n_z} \sum_{q=0}^{b} \left( \frac{n}{q} \right) \left( \frac{1-\Pi_j}{m_j-1} \right)^q \right).
\]

On the other hand, for the Problem Formulation 2, we can write the following EGF function

\[
F_1(x', y') = (e^{y'} + (x' - 1)(e^{y'} - 1))^m_g
\]

with \( x' \) is the variable that denotes the urns after some content was cached from the ground. Therefore, the associated first moment can be computed as follows

\[
E\{x'\} = m_a = m_g \left( 1 - \left( 1 - \frac{1}{m_g} \right)^{n_a} \right).
\]
We remark that the urn model used to determine which discrete media consent units can be cached near the passengers is simplified since any content not sufficiently popular was not cached in the first place from the ground catalogue.

5.3 Results

In this section, we provide the numerical results based on the model described in the previous sections. As mentioned, we divide the overall caching strategy problem for the airplane moving platform in two different problems that are both analytically tractable. Both problems have been modelled with a corresponding urn-ball allocation discrete system in which units of media contents have to be stored in units of storage. We remind that the parameter $\omega$ denotes the fraction of the total satellite bandwidth that is made available. For instance, in high throughput satellites we can expect to be close to one, which is the maximum value. On the other hand, parameter $\gamma$ denotes the popularity of a content. Furthermore, in this study, we assume that the full catalogue available on the on the ground ranges interval $m \in [10^2, 10^4]$ of different titles (e.g. movies) that can be cached on-board. The quality of each media content, which is assumed to be video for simplicity, is based on values that are readily available from experimental observations. For example, for a TV series-like video quality a data volume of approximately 3 Gigabytes/hour is generated. Hence, a two-hour video content corresponds to nearly 6 Gigabytes of data. On the other hand, we assume that the overall storage capacity available in the media server amounts to 4 Terabytes for the sake of deciding the content to cache from the ground in the first place. For deciding instead which contents to cache closer to the passengers, we assume that additional 2 Terabytes of storage is made available on-board other than the central media server.

As explained in previous sections, the total number of balls, $n$, that can be used to decide what content to cache from the ground is computed as $n = \lceil 4 \times 10^{12}/6 \times 10^9 \rceil \approx 666$ units of storage. To carry out the calculation of the number urns $n$ (i.e. storage units) that is effectively used in the discrete urn-ball assignment problem, we assume that $p_s^{(\text{sat})} = 0.9$, the parameter $\omega = \{0.2, 0.4, 0.8\}$ and $\gamma \in [0, 1]$ is the popularity parameter that we use to show the results. Further, we assume a threshold $b=1$ in the first urn-ball assignment problem and the total number of urns (i.e. contents) corresponds to $m$, which depends on $\gamma$. It is also worth noticing that using a threshold $b>1$ mainly implies that for some low values of $\gamma$ there is no content that is worth to be cached.

The result for the first urn-ball assignment problem is shown in Figure 5-11. The figure shows, as intuitively expected, that when more satellite bandwidth is available this gives the opportunity to cache more content on-board and the main limitation is the quality of the satellite connection and the amount of storage available on-board. Also, this confirms the intuition that when more content can be cached on-board, the satellite bandwidth can be spared for other services such as Internet access.

![Figure 5-11: Average number of contents that are cached from the ground versus the popularity index $\gamma$ for different fractions of the available satellite bandwidth $\omega$](image_url)
In the second part of the urn-ball assignment problem in which the strategy consists of deciding whether to cache part of the content already cached from the ground closer to the passengers, we assume that the communication link between a RTP and the test receiver makes use of a QAM modulation and the useful link distance is fixed to 4m (a realistic value on-board an aircraft). The storage parameter $K$ has a value $\left\lceil \frac{2 \times 10^{12}}{6 \times 10^9} \right\rceil \approx 333$ units of storage that is used in this case to compute the number of balls $n_a$ effectively thrown into the $m_g$ urns where $m_g$ is the result of the first urn-ball assignment problem. Other numerical values were provided already in Section 5.2.2.

The result for the second urn-ball assignment problem is shown in Figure 5-12. Interestingly, we may notice that irrespective of the popularity and the fraction of the satellite bandwidth, a plateau effect occurs. This indicates that, after storing a certain amount of content from the ground, there is no incentive to additionally cache content closer to the end users beyond a certain amount. This implies also that the additional caching effort is mainly driven by the popularity of the contents.

Figure 5-12: Average number of contents that are cached close to the passengers versus the popularity index $\gamma$ and for different fractions of the satellite bandwidth $\omega$ when QAM modulation is used at the RTP
6 Conclusions

In this deliverable, we have investigated the utilization of satellite backhaul link into the 5G system for efficient content caching and multicasting. Multiple use-cases have been considered in this deliverable to understand the performance of the SaT5G network in various content delivery and multicast scenarios. Firstly, we have investigated the possibility of offline multicasting and caching along with the delivery of OTT video live channel. We have measured the performance of caching and multicasting against three KPIs, and we have been able to make some interesting observations on latency. Then, we have investigated the performance of DASH live streaming over satellite backhaul. In this strategy, we have implemented a transient segment holding methodology and measured its performance against the various quality of experience metrics in various scenarios. We observed that the system is able to deliver seamless video streaming to the end-user over the satellite backhaul in all the scenarios by applying an appropriate holding scheme. Further, we have investigated the performance of content delivery over the SaT5G network in which mobile edge network is connected to the 5G core network via two backhaul links, namely Satellite backhaul and Terrestrial backhaul. In this work, we have considered both AVC as well as SVC layered videos. All the designed strategies in this work are able to deliver enhanced quality of video viewing experience to the end-users by optimally utilising both the available links. Then, the caching strategy problem for the case of an aircraft moving platform in which the communication system leverages on 5G technology has been investigated in the fourth section. In this work, we first model the link quality of the satellite connection when a certain fraction of the total bandwidth is made available. Afterward, we modelled the quality of a test radio link that connects a passenger’s personal device to a radio transmission point. The terminology radio transmission point was adopted to generically denote a radio transmitter that can connect client devices either through a Wi-Fi connection or by means of an LTE connection when both operate in the same 5 GHz band.
7 References


[17] N. Wang, N. Nouwell, C. Ge, B. Evans, Y. Rahulan, M. Boutin, J. Desmauts, K. Liolis, C. Politis, S. Votts and others, “Satellite support for enhanced mobile broadband content delivery in 5g,” in


